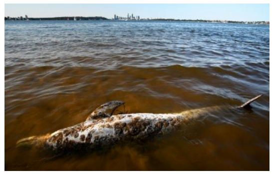
Modelling aquatic systems using the AED library









THE UNIVERSITY OF WESTERN AUSTRALIA

Centre for Water & Spatial Science

Matthew Hipsey
Centre for Water and Spatial Science
UWA School of Agriculture & Environment

GOALS FOR THIS SESSION

- Diversity and complexity in aquatic ecosystems

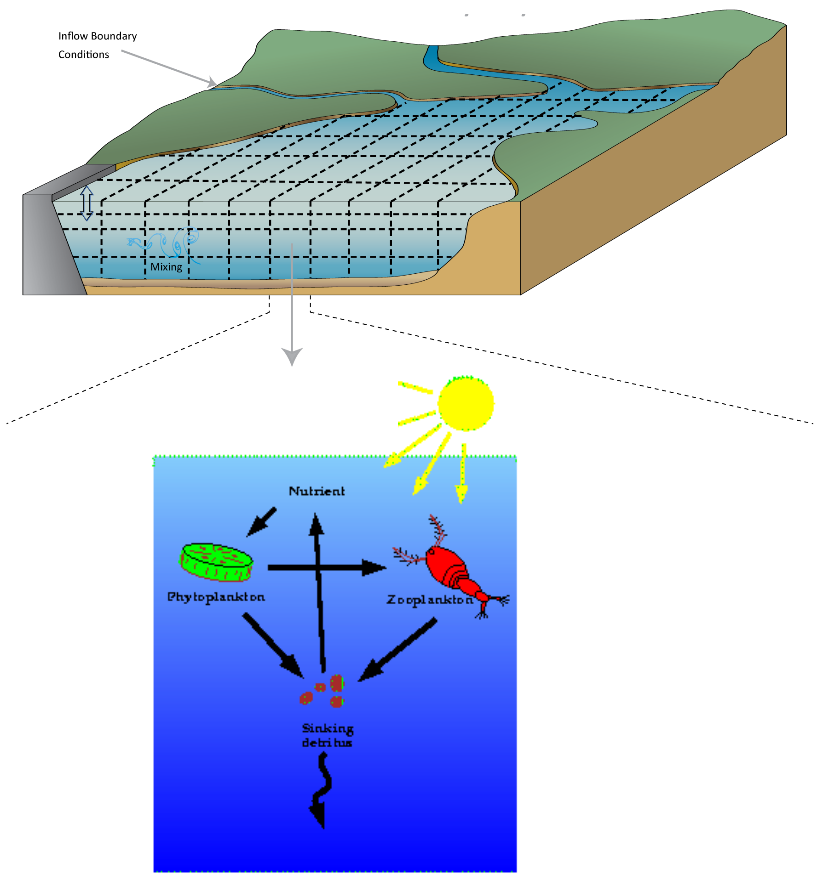
➢Need for flexibility in model frameworks

- Design of the "**AED**" library to accommodate diversity

- Run some lake simulation examples using **GLM-AED**

- Feedback and wishlist for AED and glm-py ☺. -> Follow up options

BASIC APPROACH TO AQUATIC ECOSYSTEM MODELS (AEMs)



$$\frac{dP}{dt} = d_{np} - d_{pn} - d_{pd} - d_{pz}$$

$$\frac{dZ}{dt} = d_{pz} - d_{zd} - d_{zn}$$

$$\frac{dN}{dt} = d_{dn} + d_{zn} - d_{np}$$

$$\frac{dD}{dt} = d_{pn} + d_{pd} + d_{zd} - d_{dn}$$

GENERAL APPROACH:

solve transport model

+

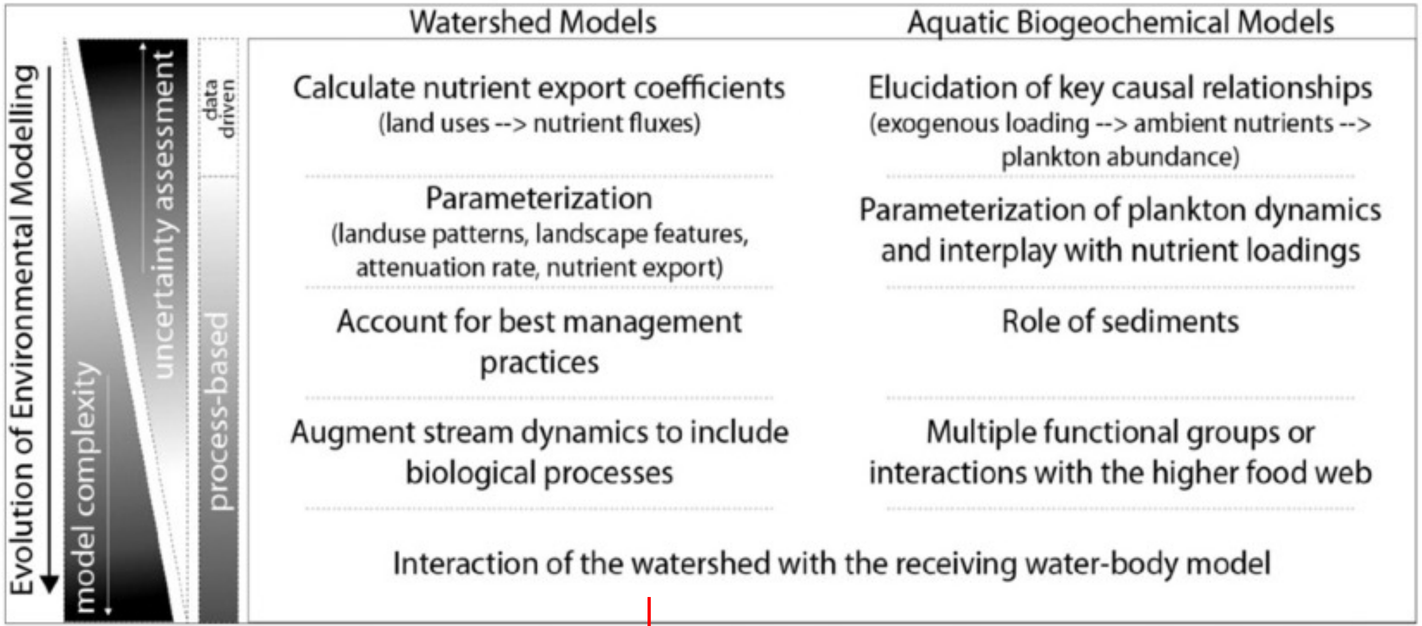
solve water quality kinetic reactions and any biological interactions in each cell

… repeat…

EVOLUTION OF WATER QUALITY MODELLING

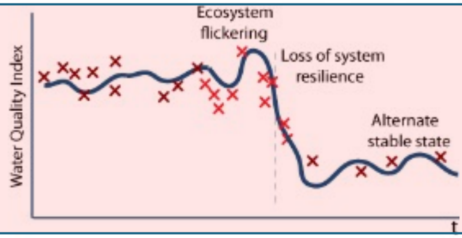
G.B. Arhonditsis et al. / Journal of Great Lakes Research 40 Supplement 3 (2014) 1–7

PAST



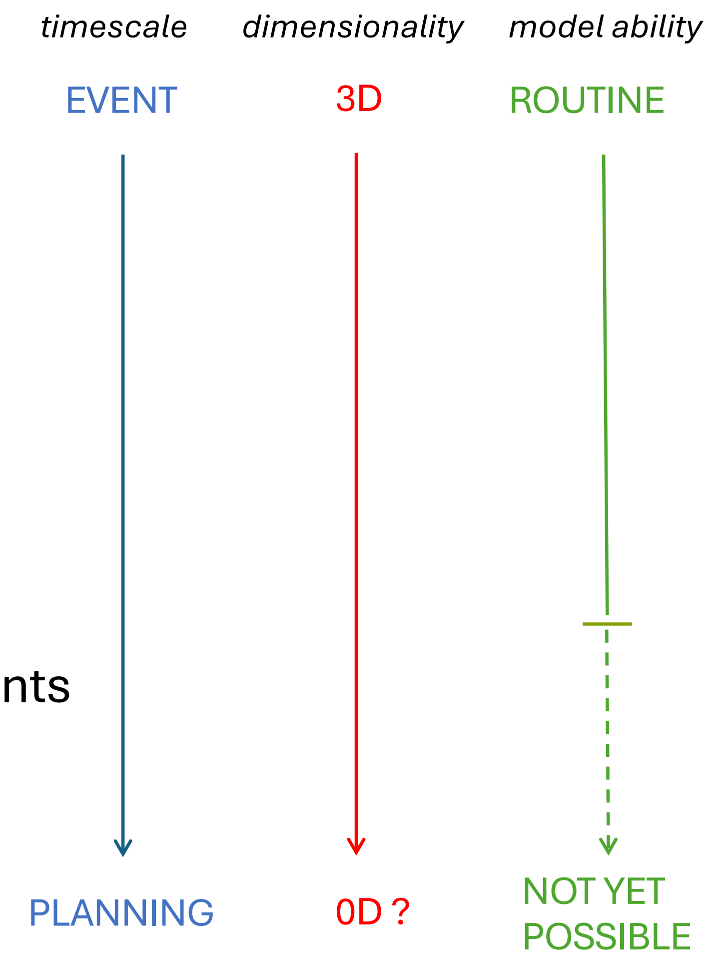
FUTURE

Integrating ML & process models
Assessing multiple ecosystem services
and system resilience

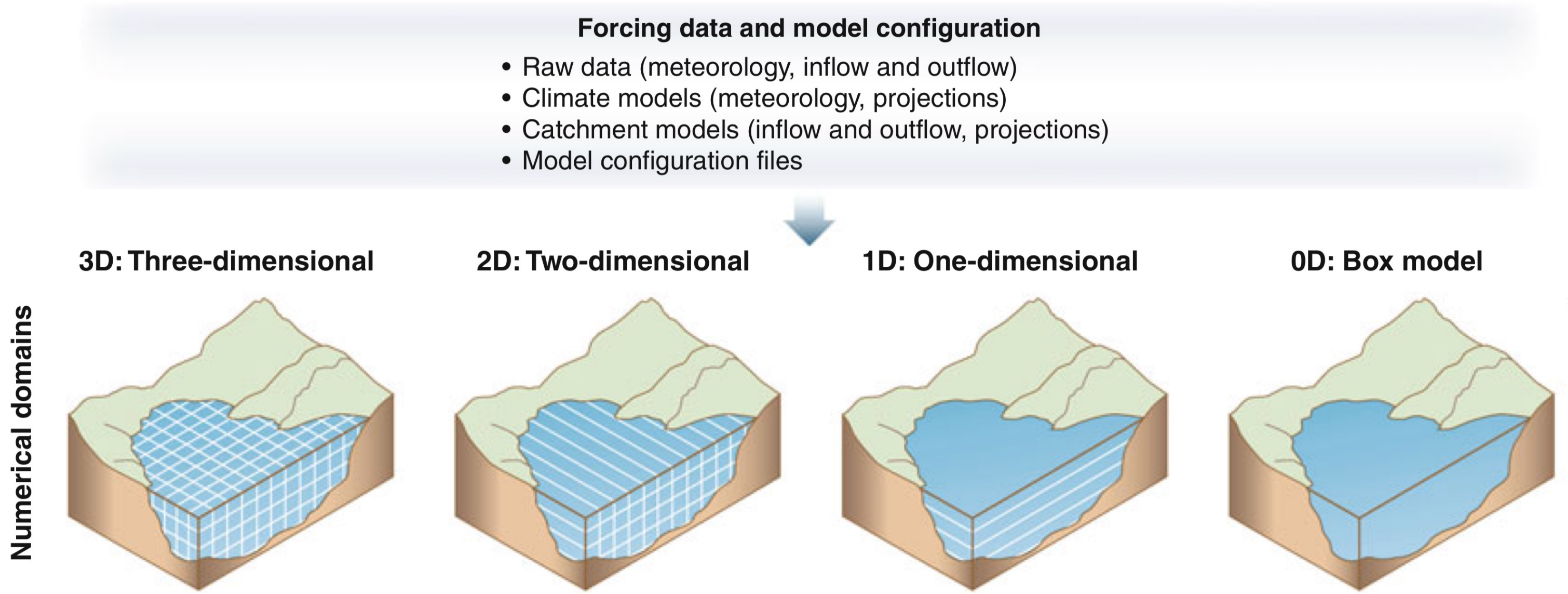


SPECTRUM OF AQUATIC ECOSYSTEM MODELLING

1. Altered hydrology & flushing
2. Hypoxia
3. Nutrients
4. Algal blooms
5. Contaminants
6. Multi-system connectivity;
   water regime management
7. System regime shifts, tipping points
8. System "*health*" and
   ecosystem service provision

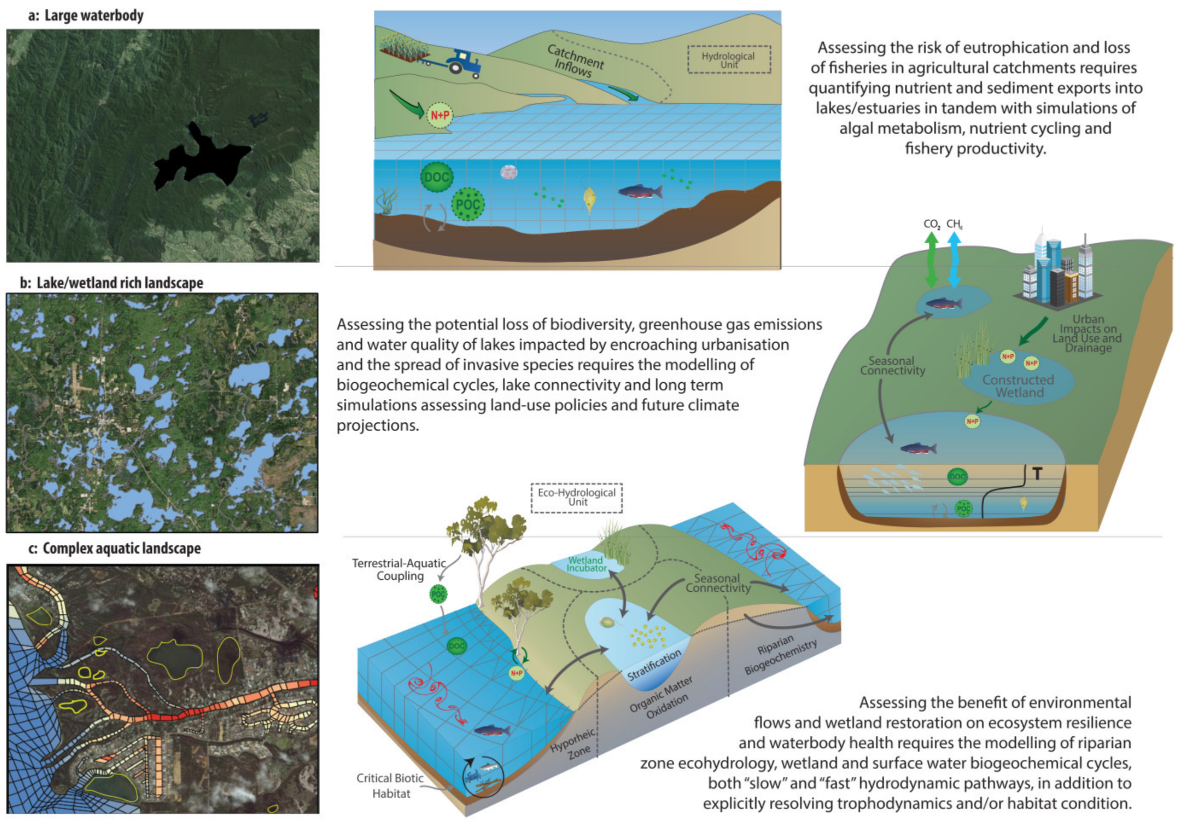


DIVERSE ECO-HYDROLOGICAL CONTEXTS

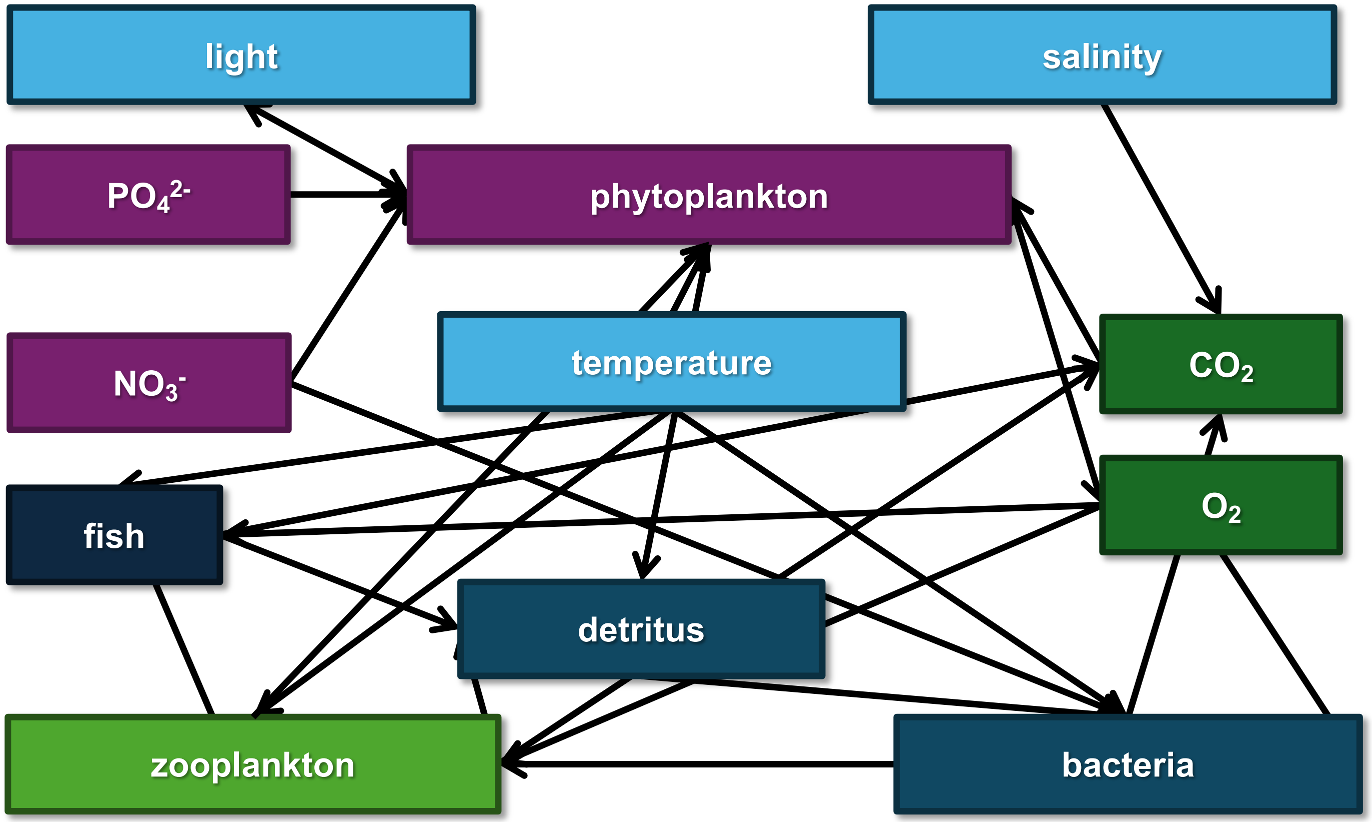


Trolle et al (2012)

DIVERSE ECO-HYDROLOGICAL CONTEXTS



HOW MUCH COMPLEXITY IS ENOUGH?

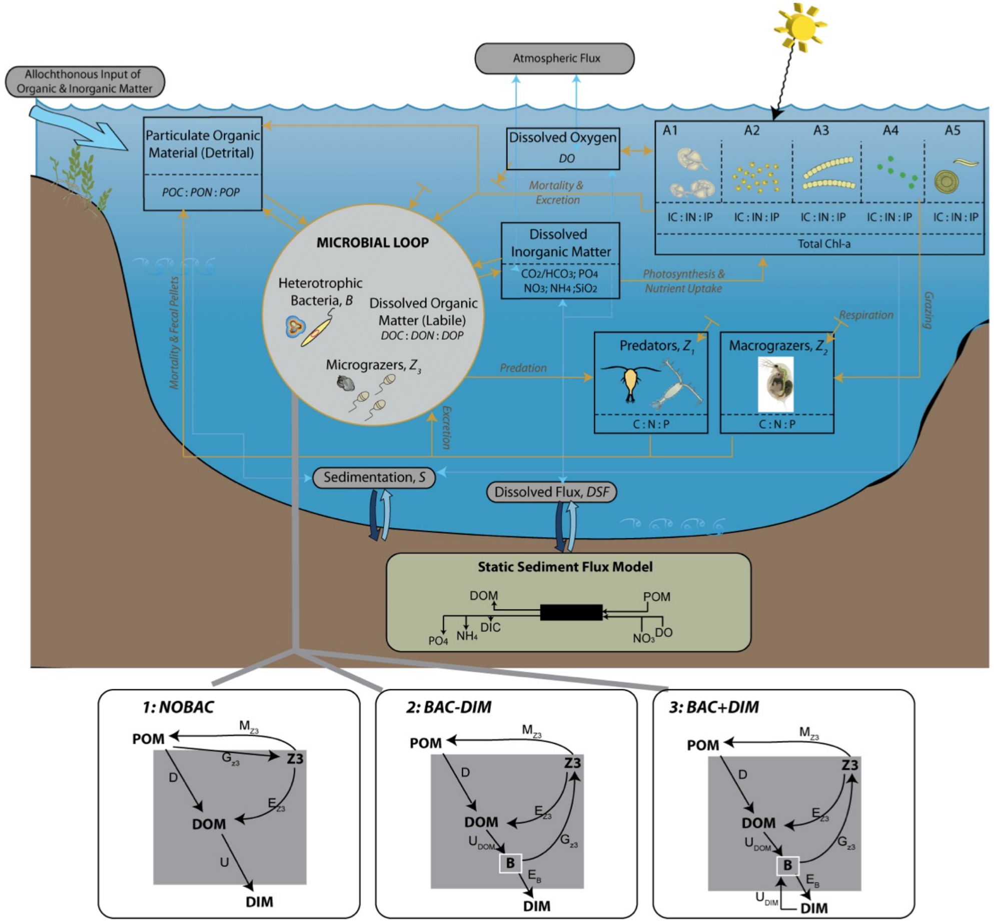


MODEL STRUCTURAL UNCERTAINTY

EXAMPLE:

How does OM recycling in the microbial loop impact nutrient flow ?

Decisions about model ”structure” matter …



Li et al. 2014

MODEL STRUCTURAL UNCERTAINTY

Including bacteria as
an intermediate between
POM and nutrients
changes subsequent nutrient
limitation of phytoplankton
groups.

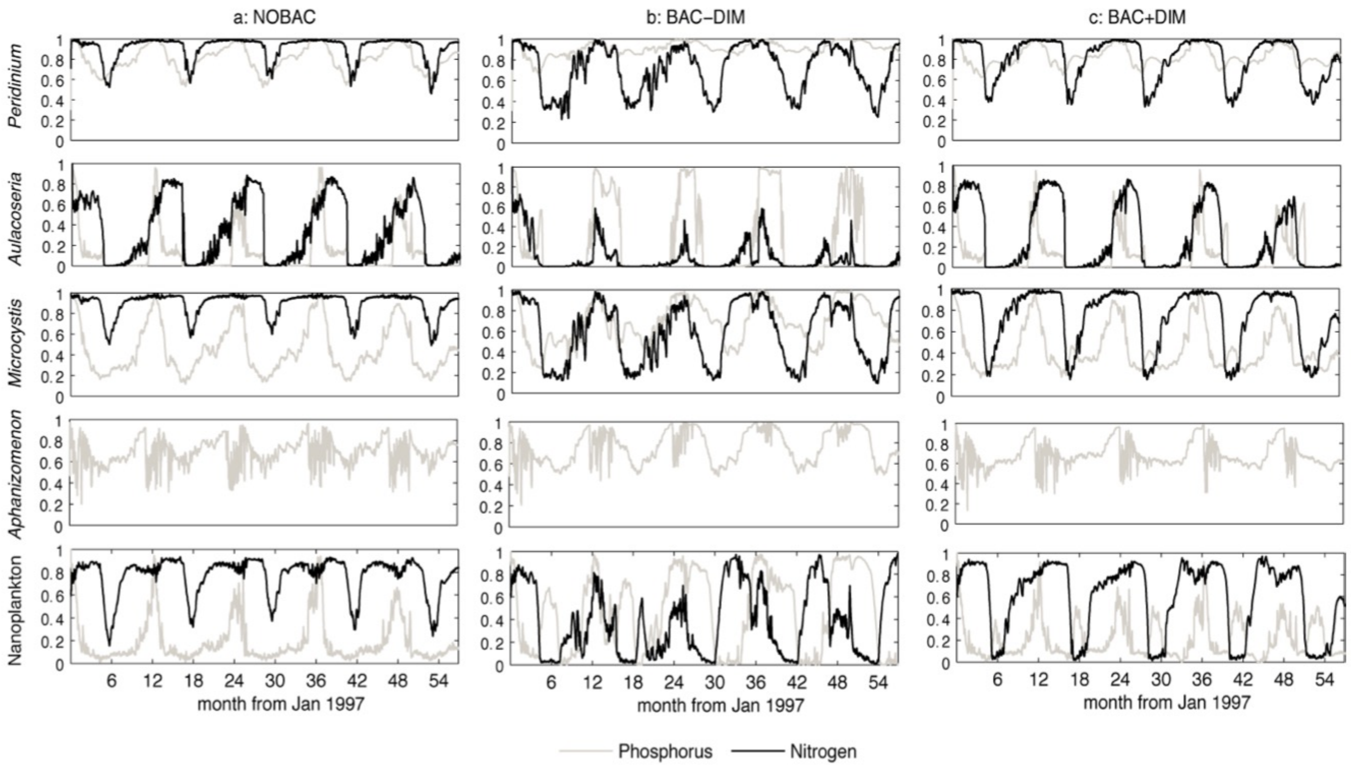


Figure 5. Comparison of nutrient limitation functions $f_a(\mathrm{N})$ and $f_a(\mathrm{P})$ respectively for the five simulated phytoplankton groups in **a)** NOBAC, **b)** BAC − DIM and **c)** BAC + DIM.

Li et al. 2014

COMPLEXITY COMPLICATES FORECASTING

- What model structure is appropriate for my system?
- Should I have more or less variables???
- Do the process parameters need changing depending in my model structure choice?
- If I am not sure about what structure to use then, what should I use for my forecasting workflow?

SO HOW DO WE DEAL WITH SO MUCH COMPLEXITY & DIVERSITY !!?!?!?

- No single model is able to resolve physical context
- No single model is able to resolve ecological interactions

➢ Create a platform that allows scientists and managers to experiment and tailor configurations to their context. e.g. :

  - Careful consideration of scale – spatial and temporal
  - Pre-existing knowledge on key process parameterisations
  - Data availability to constrain model performance
  - What is “adequate” for given purpose

AEM MODELLERS WISHLIST

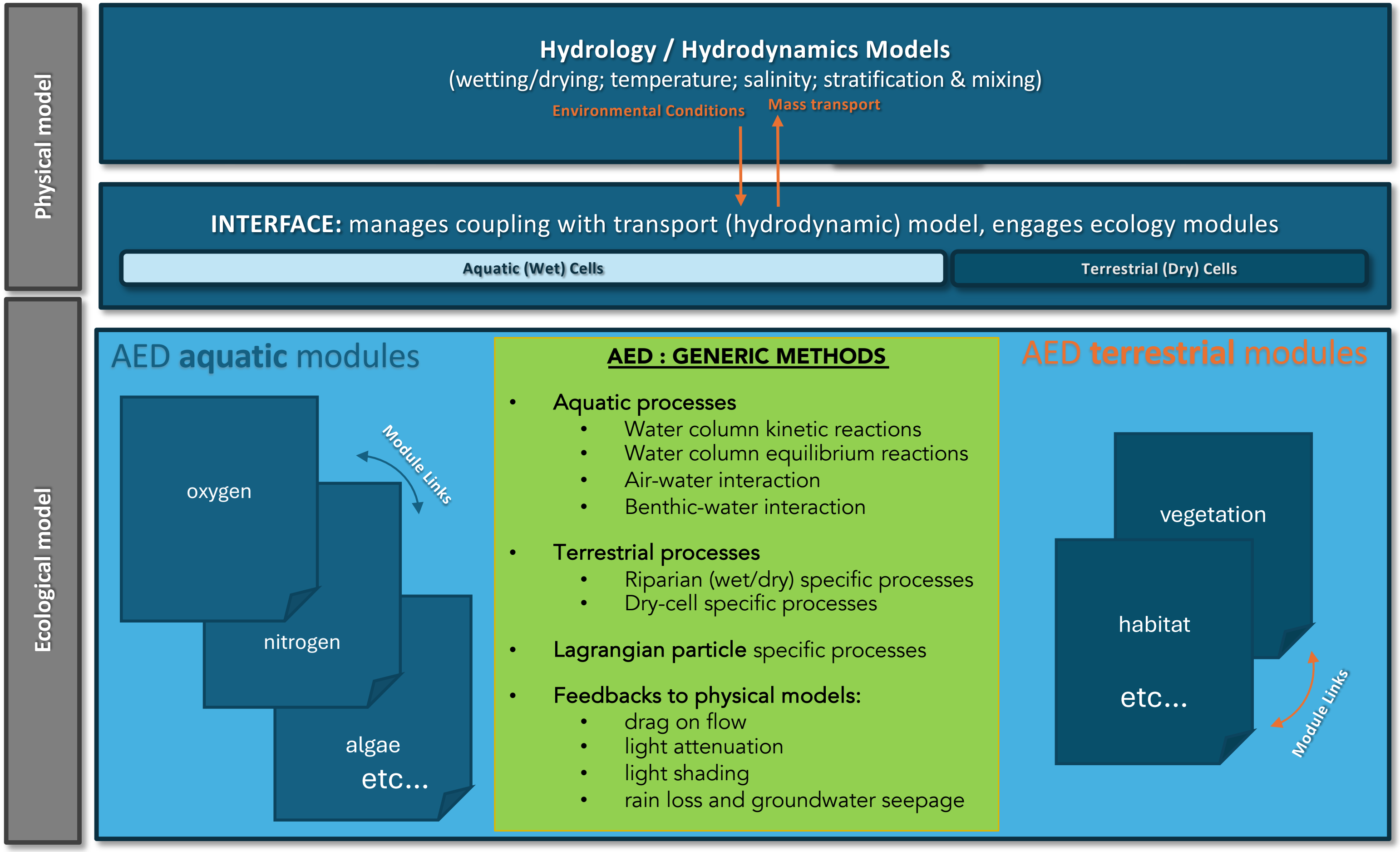
- Requirements for a flexible code-base

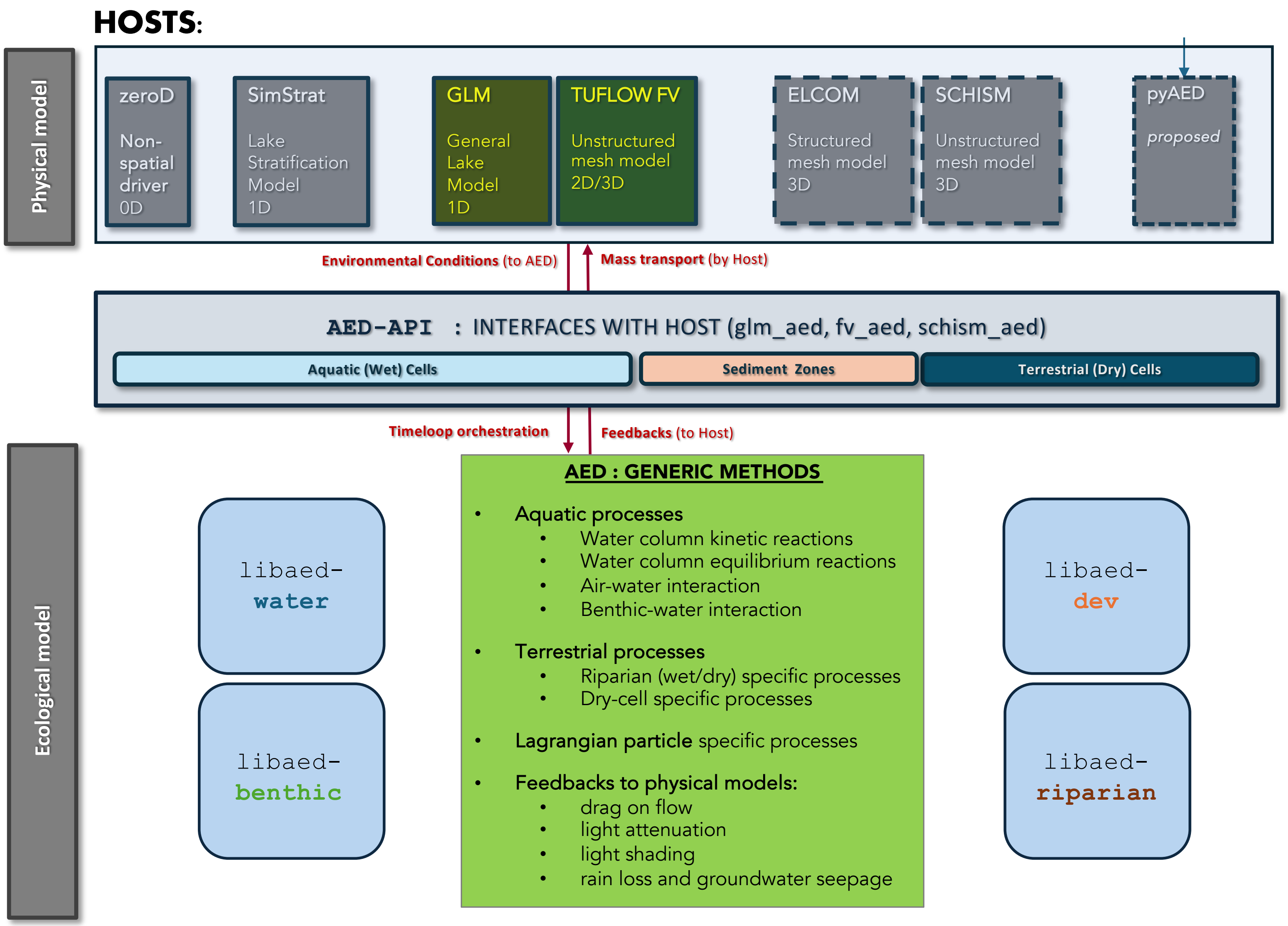
  - Multi-scale in design
    - *Flexible across spatial and temporal resolution*
  - Variable contexts
    - *Flexible to accommodate variety of external drivers (e.g. waterbody interaction with aspects of atmosphere, rivers, groundwater)*
  - Accommodate ecological model diversity
    - *Flexibility to allow simple or complex ecological conceptual models*
    - *ie. Food-web structural diversity*
  - Able to integrate with UA/DA
  - Open-source, supported and free!

Object oriented & modular model framework

AED

Capturing ecosystem & landscape diversity through flexibility in model coupling





HOSTS:
Physical model
zeroD
Non-spatial driver
0D
SimStrat
Lake Stratification Model
1D
GLM
General Lake Model
1D
TUFLOW FV
Unstructured mesh model
2D/3D
ELCOM
Structured mesh model
3D
SCHISM
Unstructured mesh model
3D
pyAED
proposed
Environmental Conditions (to AED)
Mass transport (by Host)
AED-API : INTERFACES WITH HOST (glm_aed, fv_aed, schism_aed)
Aquatic (Wet) Cells
Sediment Zones
Terrestrial (Dry) Cells
Timeloop orchestration
Feedbacks (to Host)
Ecological model
libaed-water
libaed-benthic
AED : GENERIC METHODS
Aquatic processes
Water column kinetic reactions
Water column equilibrium reactions
Air-water interaction
Benthic-water interaction
Terrestrial processes
Riparian (wet/dry) specific processes
Dry-cell specific processes
Lagrangian particle specific processes
Feedbacks to physical models:
drag on flow
light attenuation
light shading
rain loss and groundwater seepage
libaed-dev
libaed-riparian

AED & AED+ …. Aquatic Ecodynamics Modules

AED2

- **Oxygen (O_2)**
- **Suspended Sediment / Turbidity (SS)**
- **Carbon (DIC, CH_4)**
- **Nitrogen & Phosphorus (NO_x, NH_4, N_2O, PO_4, PO_4-a)**
- **Dissolved and Particulate Organic Matter (*labile & refractory*)**
- **Phytoplankton groups (*luxury N,P storage; vertical migration*)**
- **Zooplankton size groups**
- **Pathogens (*protozoa, bacteria & viruses*)**
- **Carbon, nitrogen & oxygen isotopes**
- **Geochemistry (*equilibrium; precipitation/dissolution; redox kinetics*)**
- **Sediment biogeochemistry (*early diagenesis*)**
- **Benthic plants (*macrophytes, attached & floating macroalgae*)**
- **Benthic invertebrates (*mussels*)**
- **Habitat suitability indices (*HABs, seagrass, crabs, crocs, weeds, mosquitos*)**
- **Riparian ecohydrology (*soil & vegetation*)**

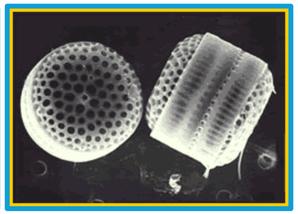
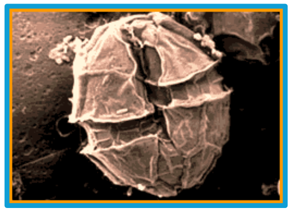
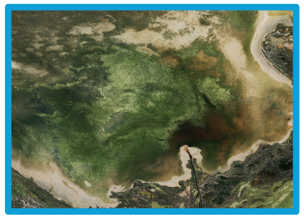
CORE

TRC NUT SDF
OXY OGM TOT
PHY ZOO

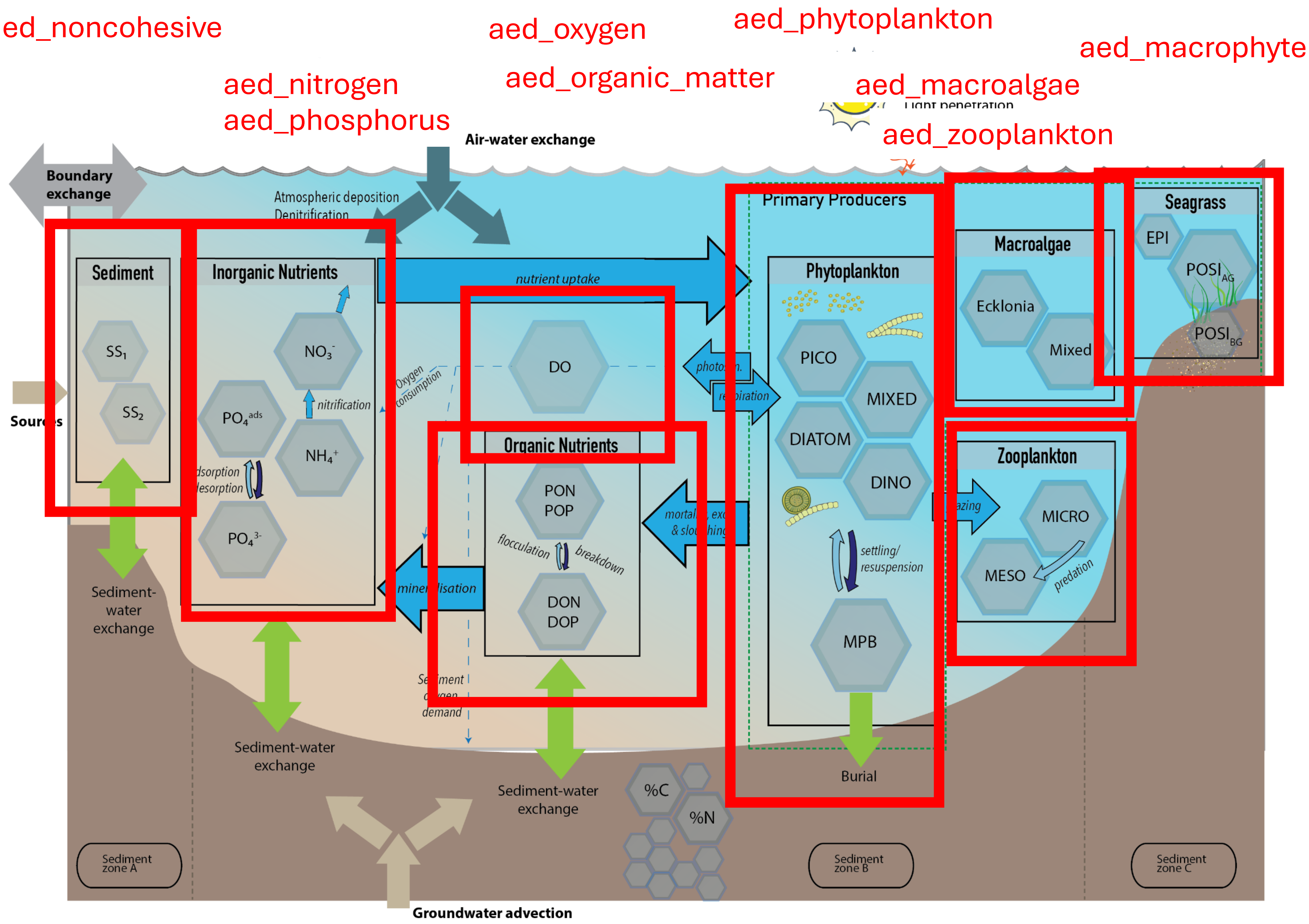
ADVANCED

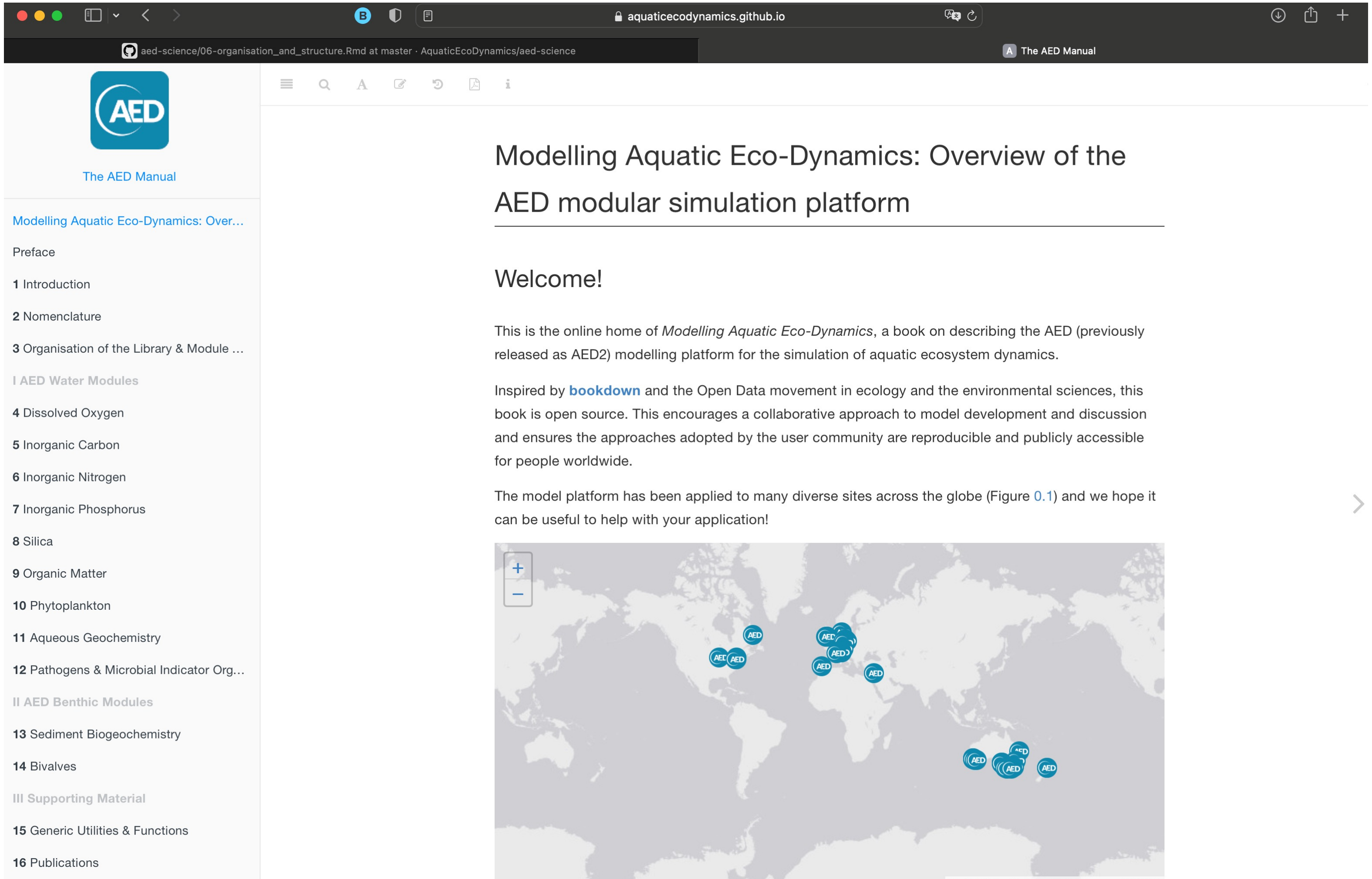
SDG GEO ISO
BIV PTH SBG
LND VEG ASS
HAB MAC MAG



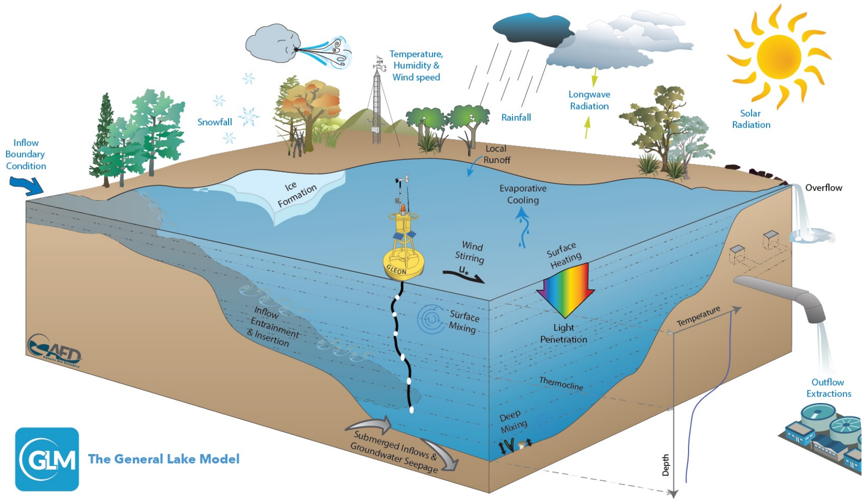
EXAMPLE AED CONCEPTUAL MODEL





https://aquaticecodynamics.github.io/aed-science/

The General Lake Model (GLM)

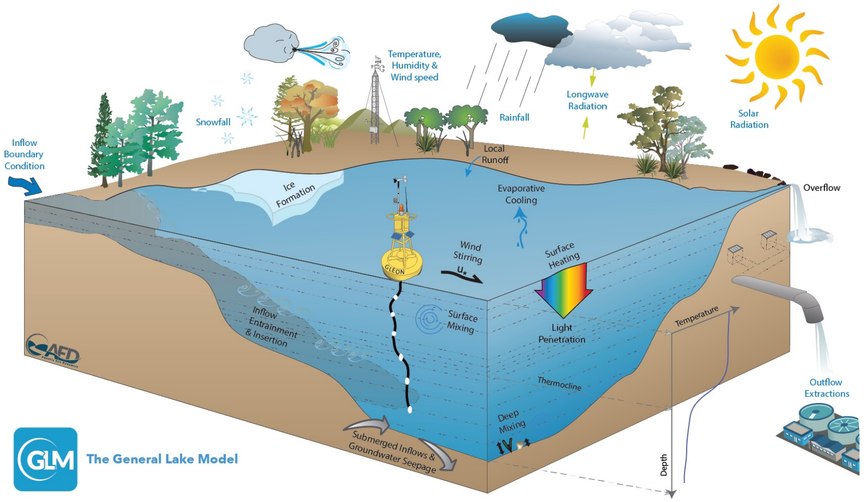


- Open-source 1D hydrodynamic model
- Water and thermal balance
- Vertical mixing & stratification
- Links with **AED** to simulate biogeochemistry

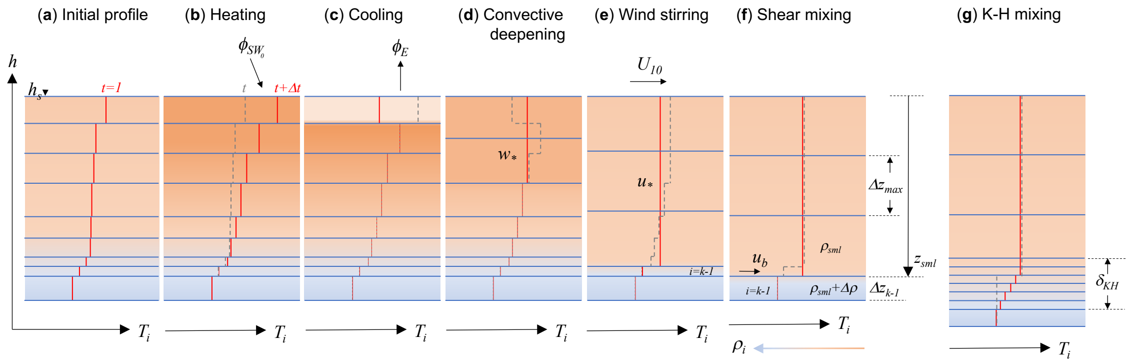
github.com/AquaticEcoDynamics/glm-aed

Hipsey, M.R., Bruce, L.C., Boon, C., Busch, B., Carey, C.C., Hamilton, D.P., Hanson, P.C., Read, J.S., de Sousa, E., Weber, M. and Winslow, L.A., 2019. **A General Lake Model (GLM 3.0) for linking with high-frequency sensor data from the Global Lake Ecological Observatory Network (GLEON)**. *Geoscientific Model Development*, *12*(1), pp.473-523.

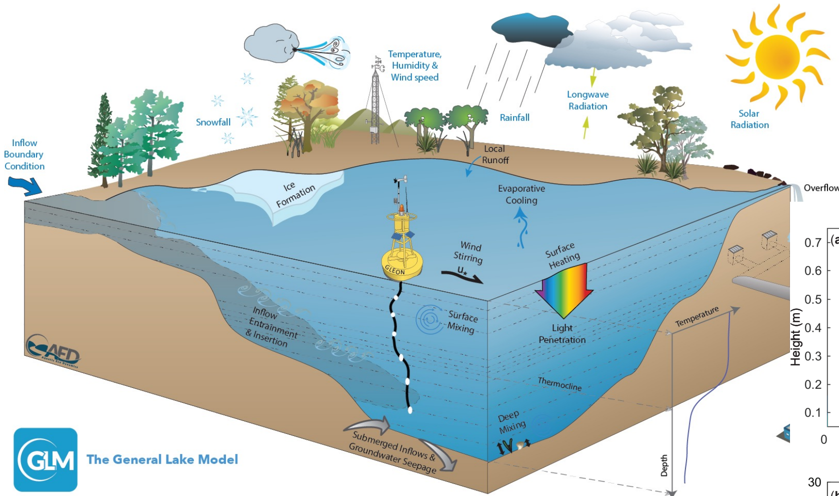
The General Lake Model (GLM)

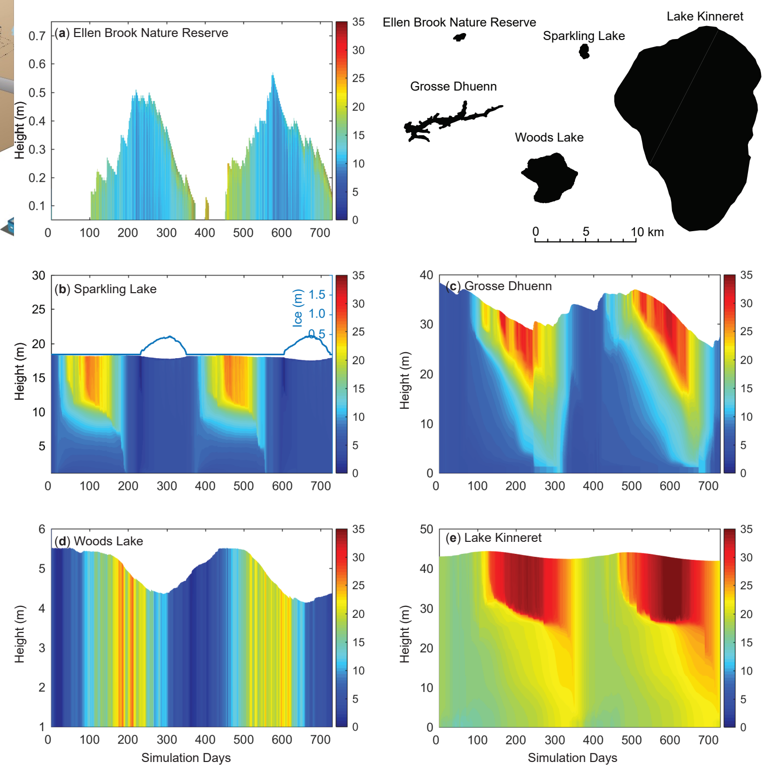


Surface mixing algorithm



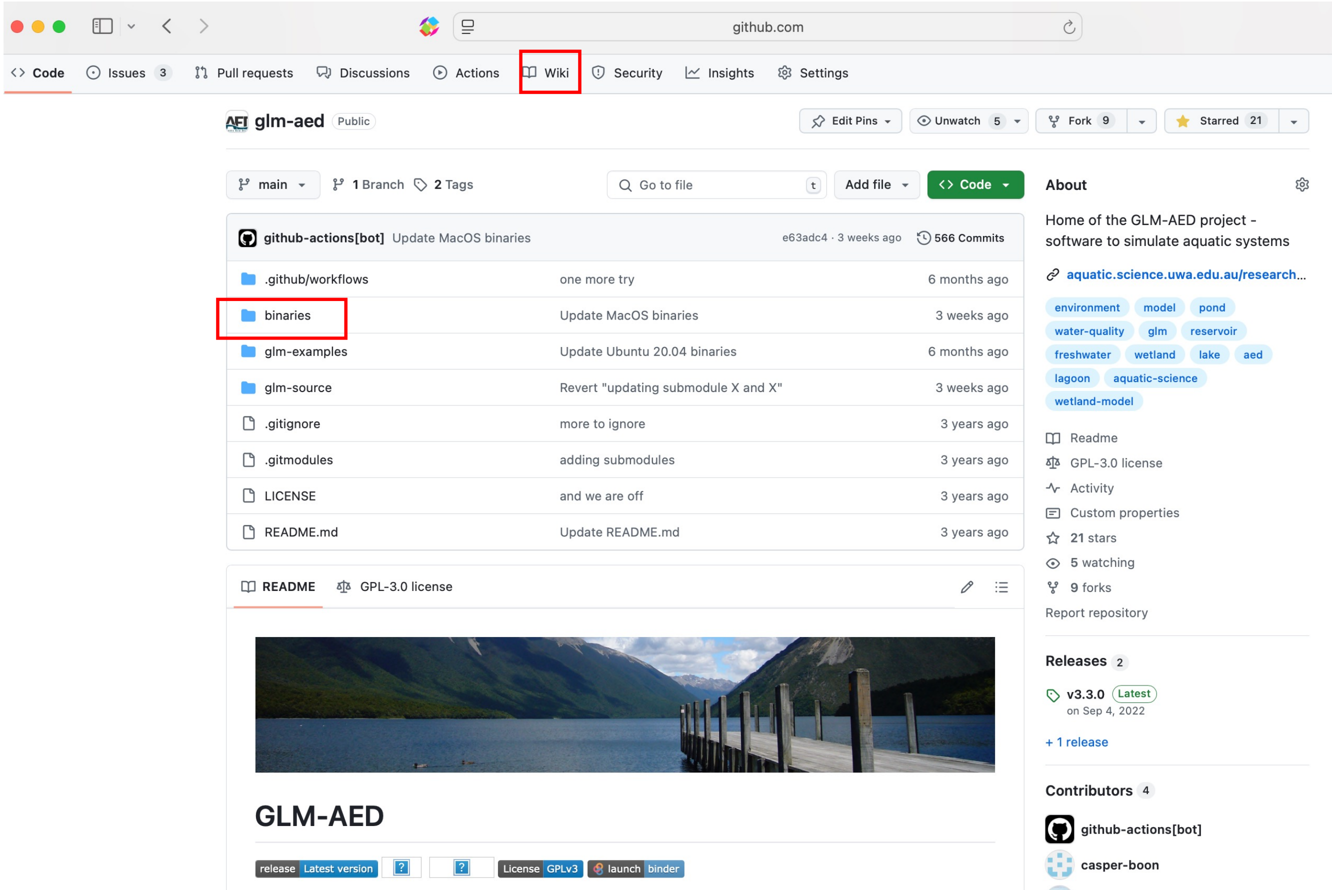
The General Lake Model (GLM)





Bruce, L.C., Frassl, M.A., Arhonditsis, G.B., Gal, G., Hamilton, D.P., Hanson, P.C., Hetherington, A.L., Melack, J.M., Read, J.S., Rinke, K., Rigosi, A. et al, 2018. **A multi-lake comparative analysis of the General Lake Model (GLM): Stress-testing across a global observatory network**. *Environmental Modelling & Software*, *102*, pp.274-291.

GLM-AED RESOURCES



LATEST:
https://github.com/AquaticEcoDynamics/glm-aed/tree/main/binaries

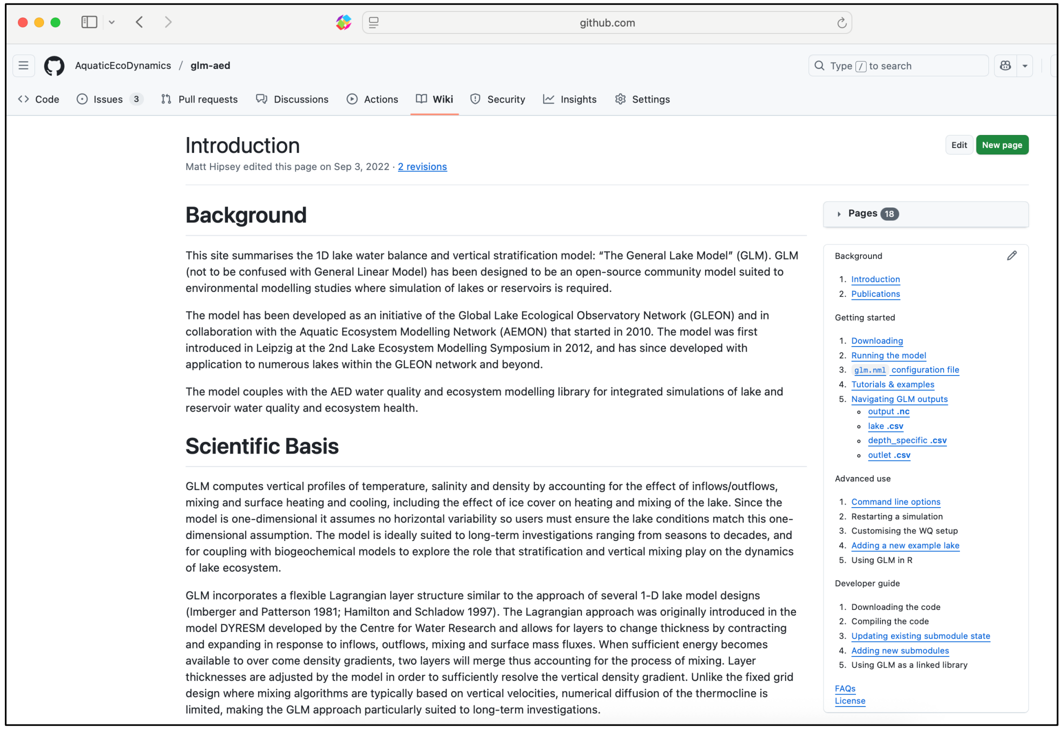
ARCHIVE:
https://github.com/AquaticEcoDynamics/releases

CURRENT:
3.3.3

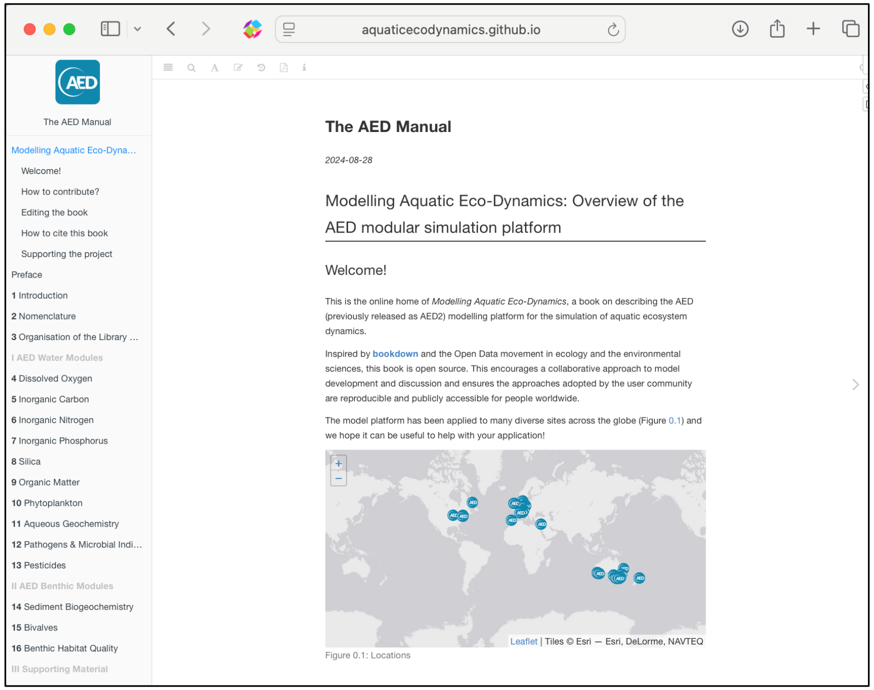
DEV (new AED API):
3.9.X

GLM-AED RESOURCES

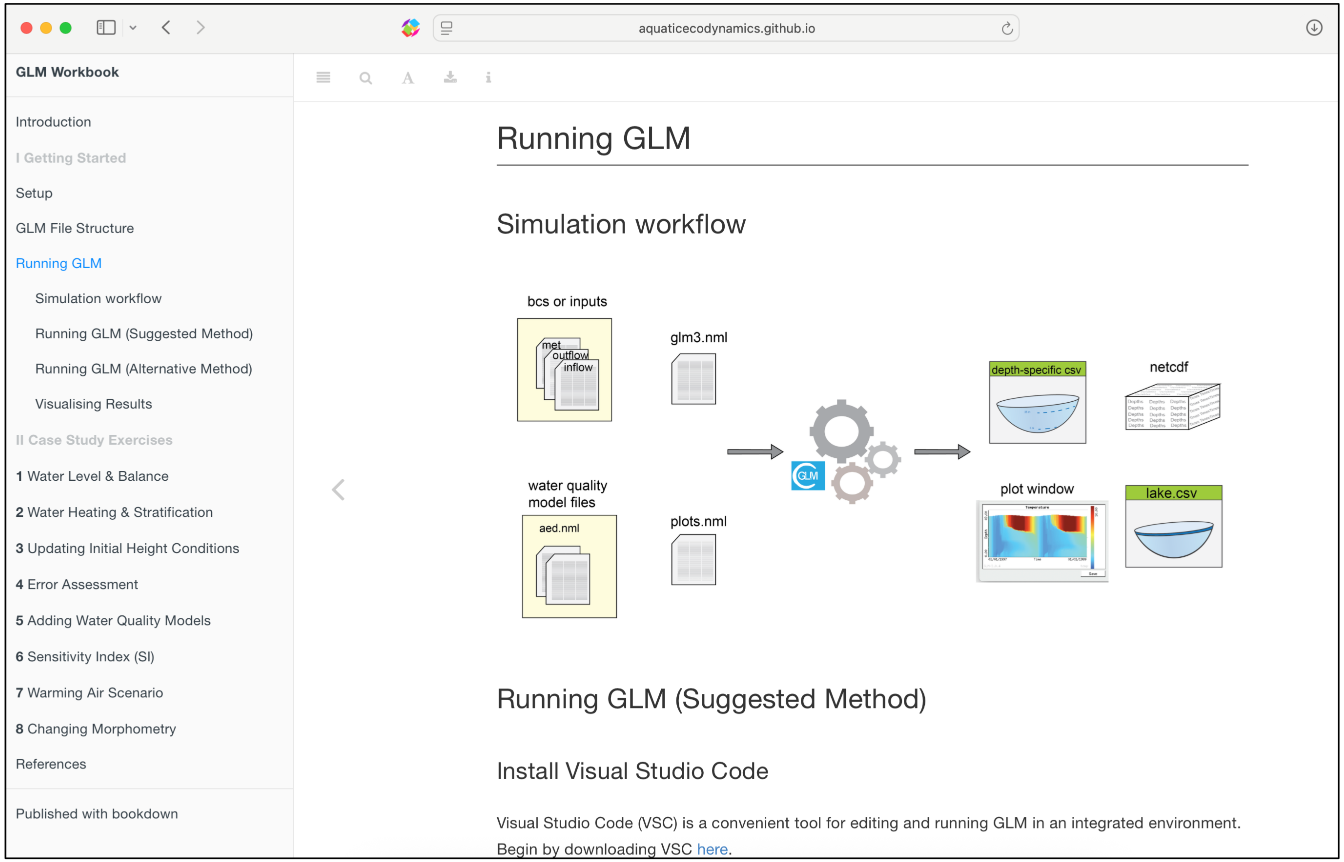
GLM Wiki

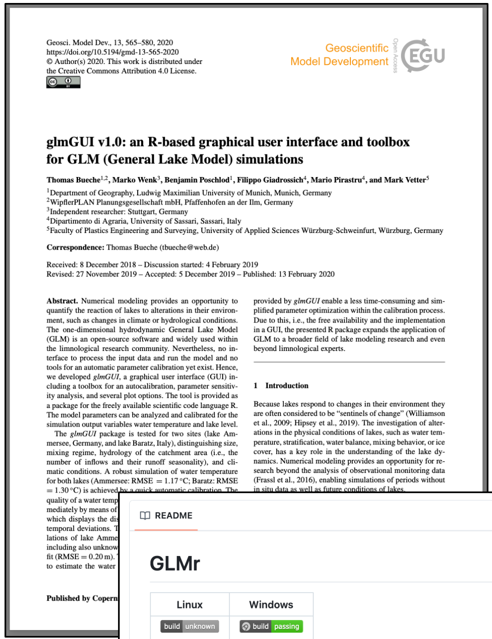


AED Manual



GLM-WORKBOOK




Geosci. Model Dev., 13, 565–580, 2020
https://doi.org/10.5194/gmd-13-565-2020
© Author(s) 2020. This work is distributed under the Creative Commons Attribution 4.0 License.

Geoscientific Model Development — Open Access — EGU

glmGUI v1.0: an R-based graphical user interface and toolbox for GLM (General Lake Model) simulations

Thomas Bueche[1,2], Marko Wenk[3], Benjamin Poschlod[1], Filippo Giadrossich[4], Mario Pirastru[4], and Mark Vetter[5]

[1]Department of Geography, Ludwig Maximilian University of Munich, Munich, Germany
[2]WipflerPLAN Planungsgesellschaft mbH, Pfaffenhofen an der Ilm, Germany
[3]Independent researcher: Stuttgart, Germany
[4]Dipartimento di Agraria, University of Sassari, Sassari, Italy
[5]Faculty of Plastics Engineering and Surveying, University of Applied Sciences Würzburg-Schweinfurt, Würzburg, Germany

Correspondence: Thomas Bueche (tbueche@web.de)

Received: 8 December 2018 – Discussion started: 4 February 2019
Revised: 27 November 2019 – Accepted: 5 December 2019 – Published: 13 February 2020

Abstract. Numerical modeling provides an opportunity to quantify the reaction of lakes to alterations in their environment, such as changes in climate or hydrological conditions. The one-dimensional hydrodynamic General Lake Model (GLM) is an open-source software and widely used within the limnological research community. Nevertheless, no interface to process the input data and run the model and no tools for an automatic parameter calibration yet exist. Hence, we developed *glmGUI*, a graphical user interface (GUI) including a toolbox for an autocalibration, parameter sensitivity analysis, and several plot options. The tool is provided as a package for the freely available scientific code language R. The model parameters can be analyzed and calibrated for the simulation output variables water temperature and lake level.

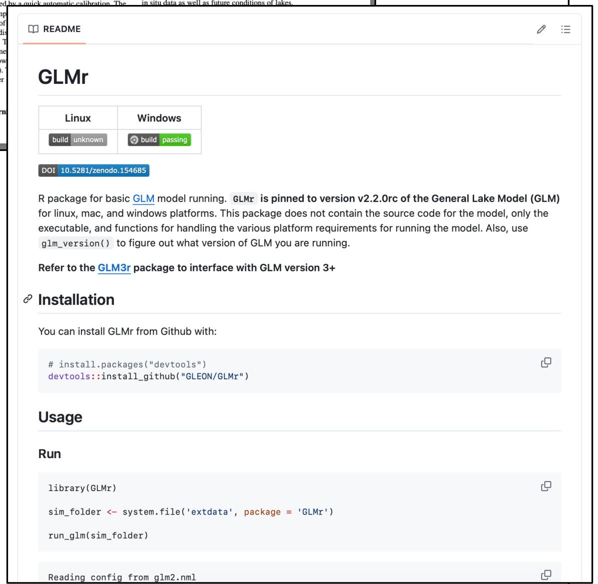
The *glmGUI* package is tested for two sites (lake Ammersee, Germany, and lake Baratz, Italy), distinguishing size, mixing regime, hydrology of the catchment area (i.e., the number of inflows and their runoff seasonality), and climatic conditions. A robust simulation of water temperature for both lakes (Ammersee: RMSE = 1.17 °C; Baratz: RMSE = 1.30 °C) is achieved by a quick automatic calibration. The quality of a water temp… mediately by means of… which displays the dis… temporal deviations. T… lations of lake Amme… including also unknow… fit (RMSE = 0.20 m). … to estimate the water …

… provided by *glmGUI* enable a less time-consuming and simplified parameter optimization within the calibration process. Due to this, i.e., the free availability and the implementation in a GUI, the presented R package expands the application of GLM to a broader field of lake modeling research and even beyond limnological experts.

1 Introduction

Because lakes respond to changes in their environment they are often considered to be "sentinels of change" (Williamson et al., 2009; Hipsey et al., 2019). The investigation of alterations in the physical conditions of lakes, such as water temperature, stratification, water balance, mixing behavior, or ice cover, has a key role in the understanding of the lake dynamics. Numerical modeling provides an opportunity for research beyond the analysis of observational monitoring data (Frassl et al., 2016), enabling simulations of periods without in situ data as well as future conditions of lakes.

Published by Copern…


README

GLMr

Linux	Windows
build unknown	build passing

DOI 10.5281/zenodo.154685

R package for basic GLM model running. `GLMr` **is pinned to version v2.2.0rc of the General Lake Model (GLM)** for linux, mac, and windows platforms. This package does not contain the source code for the model, only the executable, and functions for handling the various platform requirements for running the model. Also, use `glm_version()` to figure out what version of GLM you are running.

Refer to the GLM3r package to interface with GLM version 3+

Installation

You can install GLMr from Github with:

```
# install.packages("devtools")
devtools::install_github("GLEON/GLMr")
```

Usage

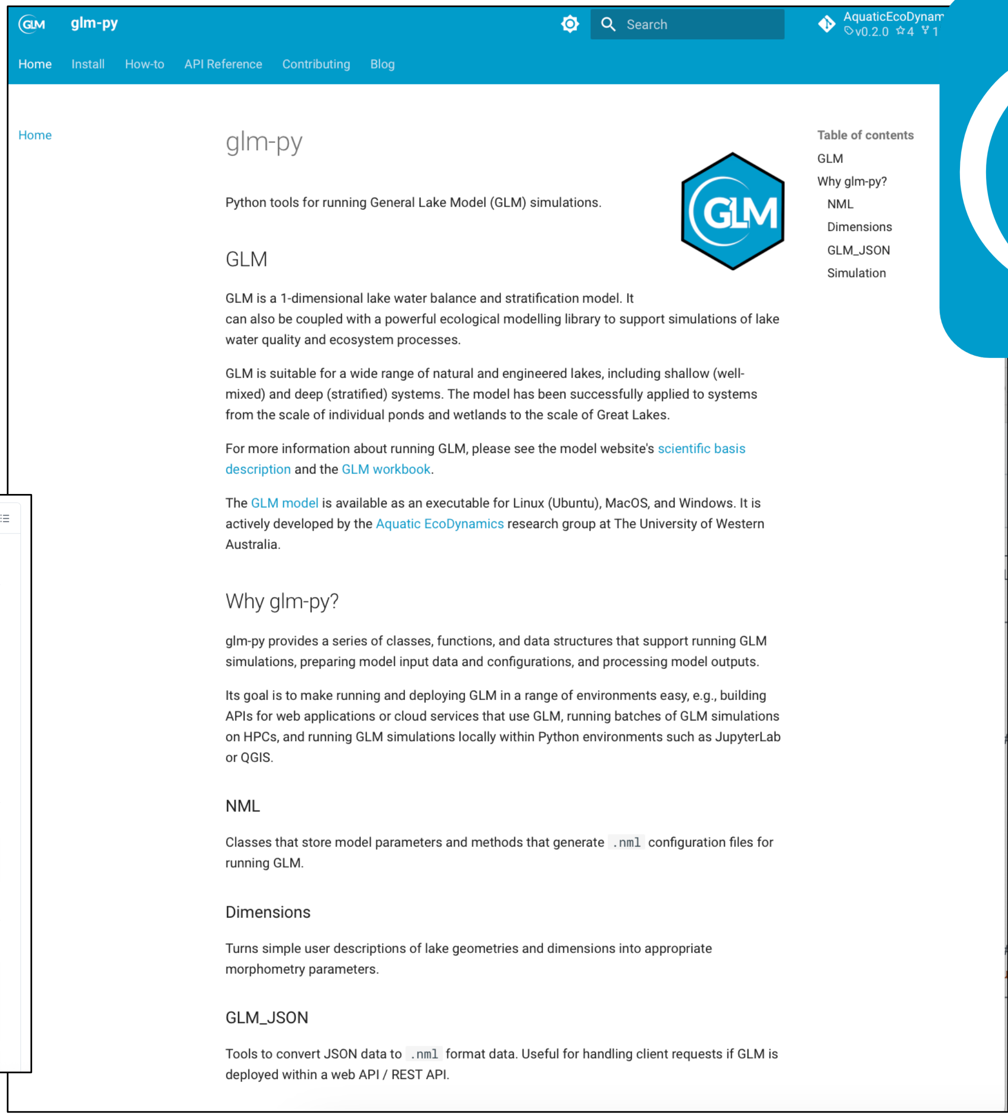
Run

```
library(GLMr)

sim_folder <- system.file('extdata', package = 'GLMr')

run_glm(sim_folder)
```

```
Reading config from glm2.nml
```


glm-py

Search

AquaticEcoDynam… v0.2.0 ☆4 ⑂1

Home Install How-to API Reference Contributing Blog

Home

glm-py

Python tools for running General Lake Model (GLM) simulations.

GLM

GLM is a 1-dimensional lake water balance and stratification model. It can also be coupled with a powerful ecological modelling library to support simulations of lake water quality and ecosystem processes.

GLM is suitable for a wide range of natural and engineered lakes, including shallow (well-mixed) and deep (stratified) systems. The model has been successfully applied to systems from the scale of individual ponds and wetlands to the scale of Great Lakes.

For more information about running GLM, please see the model website's scientific basis description and the GLM workbook.

The GLM model is available as an executable for Linux (Ubuntu), MacOS, and Windows. It is actively developed by the Aquatic EcoDynamics research group at The University of Western Australia.

Why glm-py?

glm-py provides a series of classes, functions, and data structures that support running GLM simulations, preparing model input data and configurations, and processing model outputs.

Its goal is to make running and deploying GLM in a range of environments easy, e.g., building APIs for web applications or cloud services that use GLM, running batches of GLM simulations on HPCs, and running GLM simulations locally within Python environments such as JupyterLab or QGIS.

NML

Classes that store model parameters and methods that generate `.nml` configuration files for running GLM.

Dimensions

Turns simple user descriptions of lake geometries and dimensions into appropriate morphometry parameters.

GLM_JSON

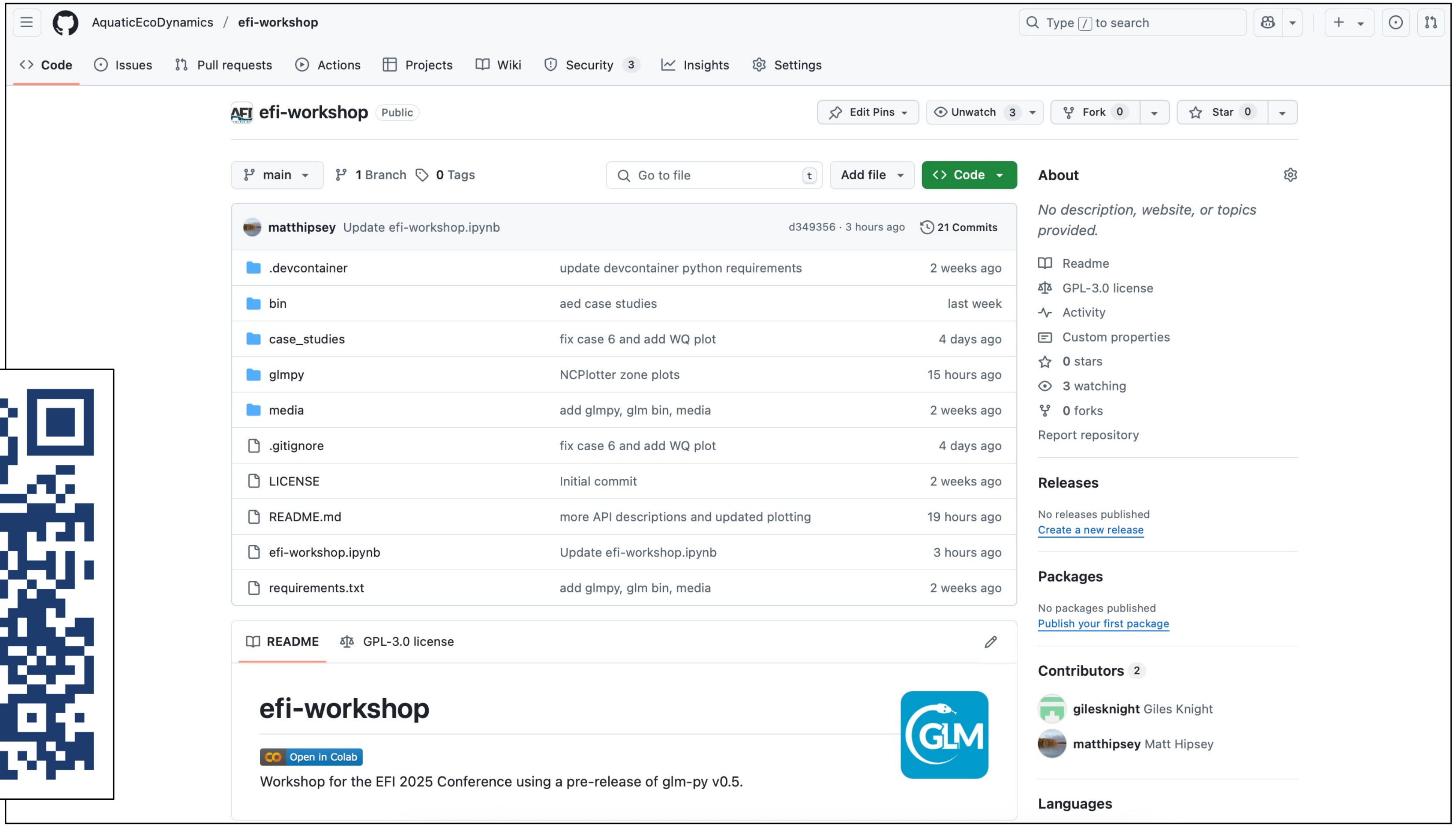
Tools to convert JSON data to `.nml` format data. Useful for handling client requests if GLM is deployed within a web API / REST API.

Table of contents

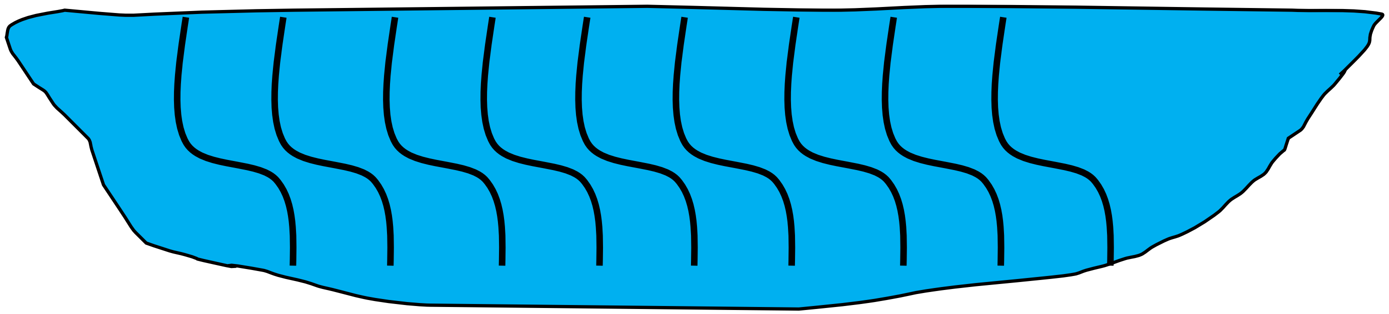
glm-py

Developer:
Giles Knight

EFI WORKSHOP NOTEBOOK



OK – LETS BUILD UP A ECOSYSTEM MODEL ....



OXYGEN

CHLA

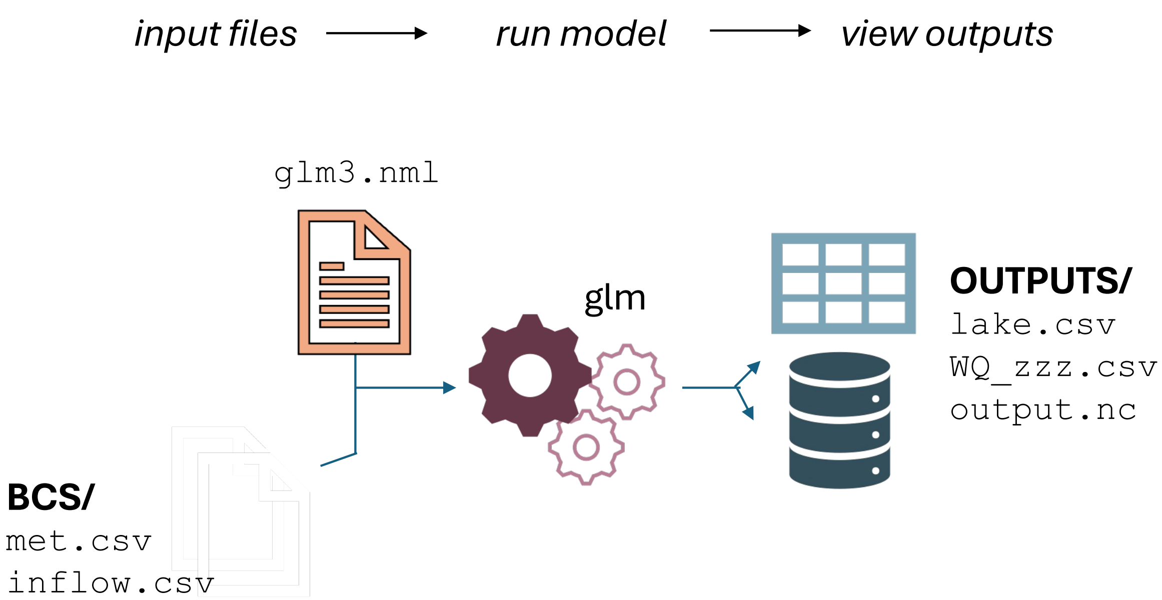
NUTRIENTS

PHYTOPLANKTON

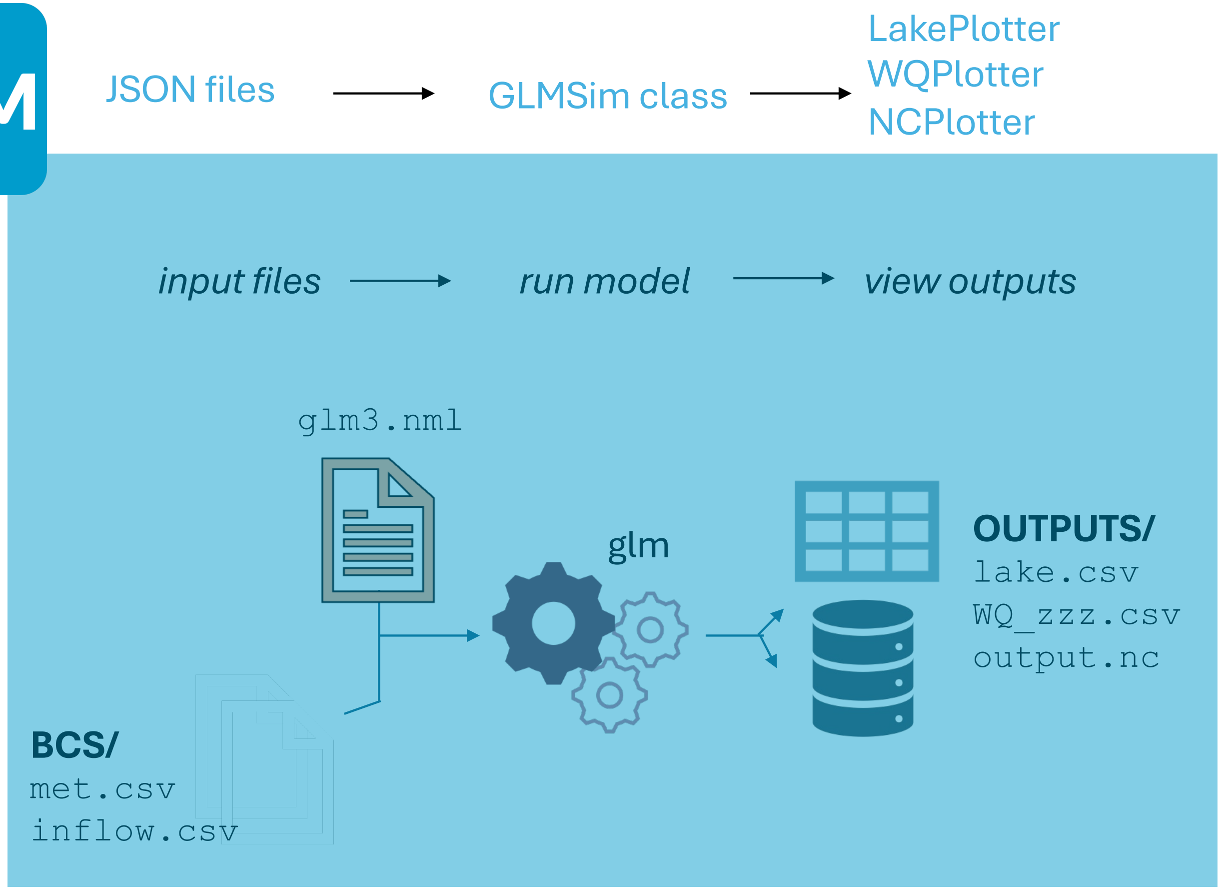
ZOOPLANKTON

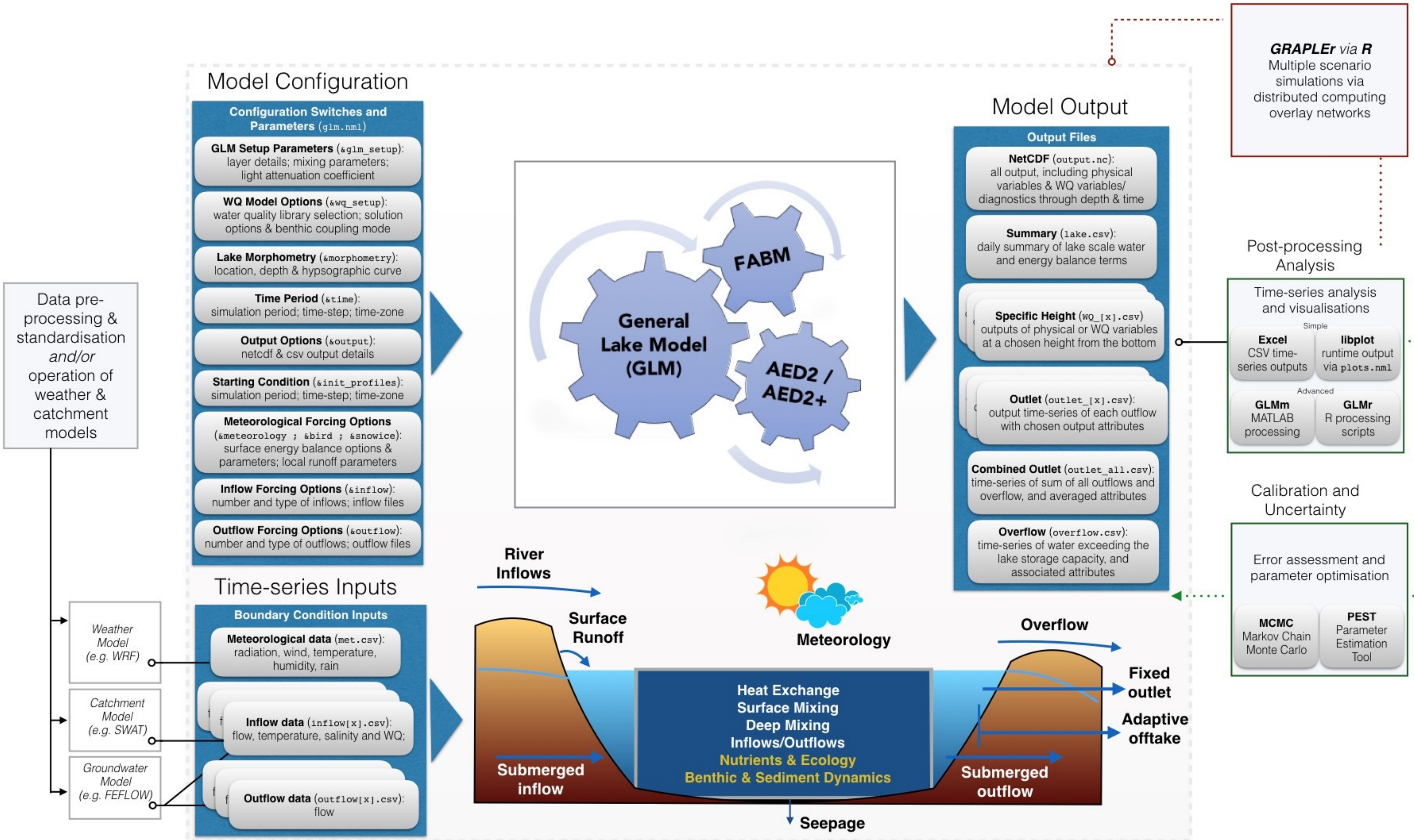
MACROPHYTE

BASIC WORKFLOW



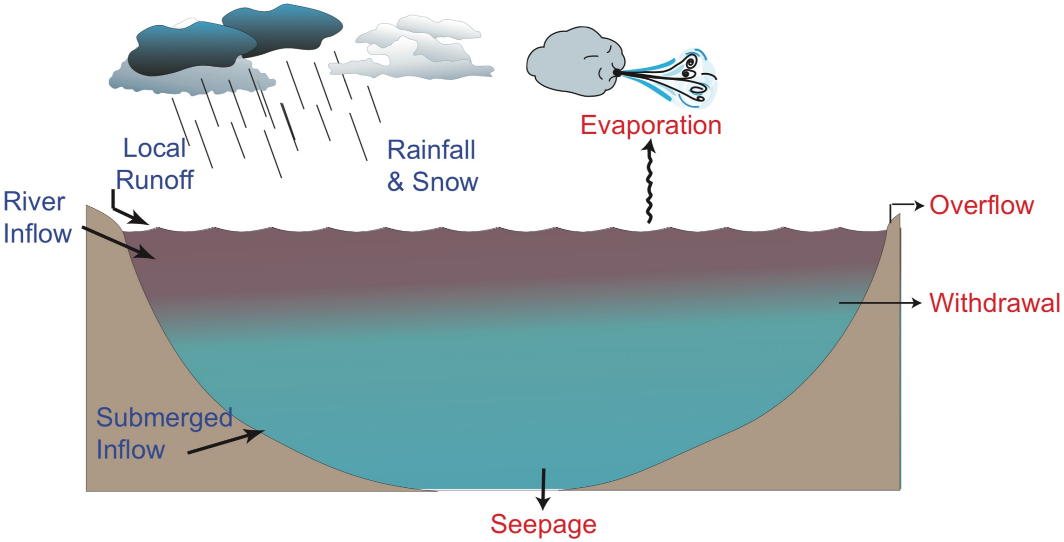
PYTHON WORKFLOW





Model Configuration
Configuration Switches and Parameters (glm.nml)
GLM Setup Parameters (&glm_setup): layer details; mixing parameters; light attenuation coefficient
WQ Model Options (&wq_setup): water quality library selection; solution options & benthic coupling mode
Lake Morphometry (&morphometry): location, depth & hypsographic curve
Time Period (&time): simulation period; time-step; time-zone
Output Options (&output): netcdf & csv output details
Starting Condition (&init_profiles): simulation period; time-step; time-zone
Meteorological Forcing Options (&meteorology ; &bird ; &snowice): surface energy balance options & parameters; local runoff parameters
Inflow Forcing Options (&inflow): number and type of inflows; inflow files
Outflow Forcing Options (&outflow): number and type of outflows; outflow files
General Lake Model (GLM)
FABM
AED2 / AED2+
Model Output
Output Files
NetCDF (output.nc): all output, including physical variables & WQ variables/ diagnostics through depth & time
Summary (lake.csv): daily summary of lake scale water and energy balance terms
Specific Height (WQ_[x].csv) outputs of physical or WQ variables at a chosen height from the bottom
Outlet (outlet_[x].csv): output time-series of each outflow with chosen output attributes
Combined Outlet (outlet_all.csv): time-series of sum of all outflows and overflow, and averaged attributes
Overflow (overflow.csv): time-series of water exceeding the lake storage capacity, and associated attributes
GRAPLEr via R
Multiple scenario simulations via distributed computing overlay networks
Post-processing Analysis
Time-series analysis and visualisations
Simple
Excel CSV time-series outputs
libplot runtime output via plots.nml
Advanced
GLMm MATLAB processing
GLMr R processing scripts
Calibration and Uncertainty
Error assessment and parameter optimisation
MCMC Markov Chain Monte Carlo
PEST Parameter Estimation Tool
Data pre-processing & standardisation and/or operation of weather & catchment models
Weather Model (e.g. WRF)
Catchment Model (e.g. SWAT)
Groundwater Model (e.g. FEFLOW)
Time-series Inputs
Boundary Condition Inputs
Meteorological data (met.csv): radiation, wind, temperature, humidity, rain
Inflow data (inflow[x].csv): flow, temperature, salinity and WQ;
Outflow data (outflow[x].csv): flow
River Inflows
Surface Runoff
Meteorology
Overflow
Fixed outlet
Adaptive offtake
Heat Exchange
Surface Mixing
Deep Mixing
Inflows/Outflows
Nutrients & Ecology
Benthic & Sediment Dynamics
Submerged inflow
Submerged outflow
Seepage

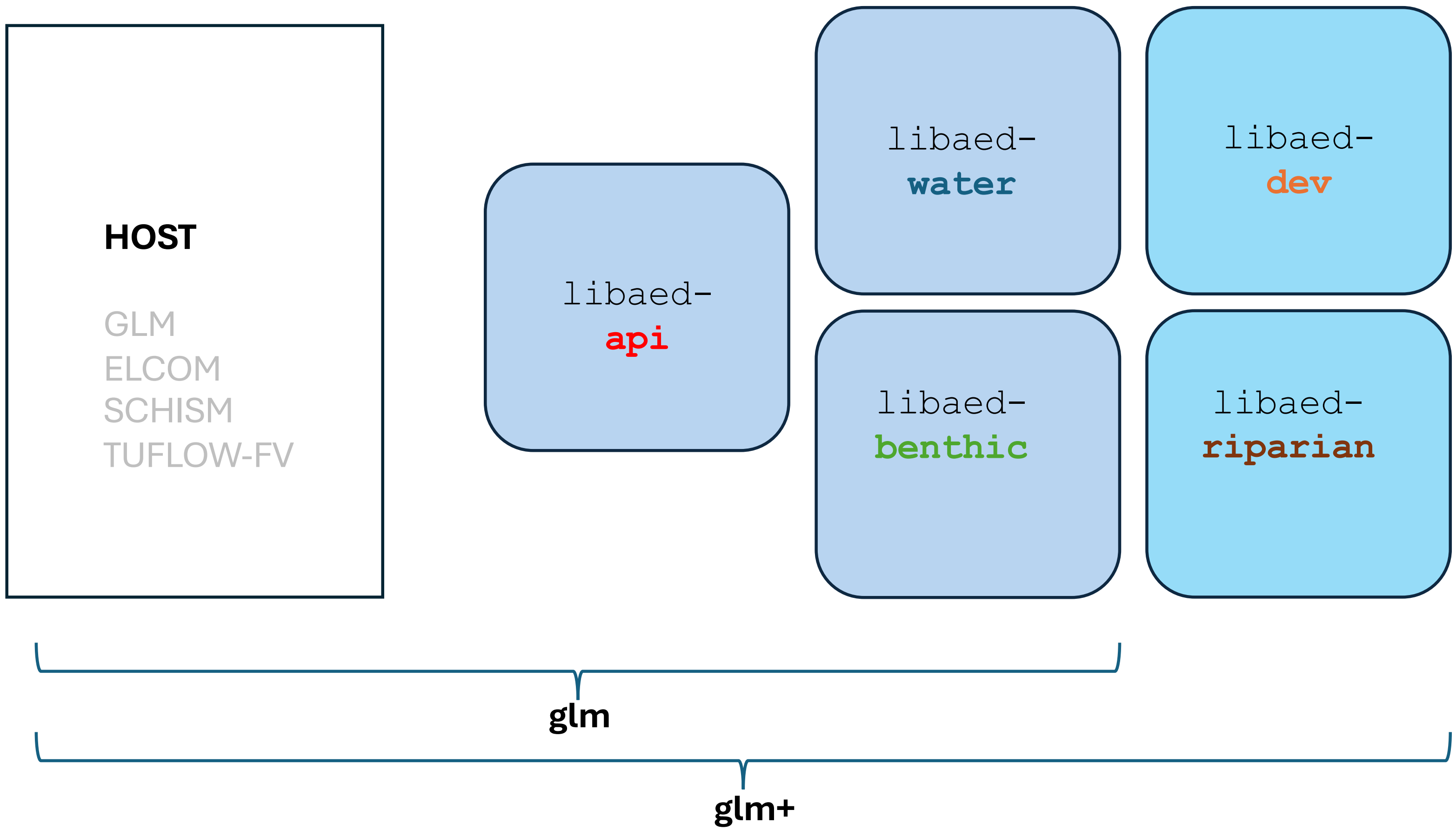
INITIAL TASK – RUN THE MODEL AND PLOT THE WATER BALANCE



$$\frac{dS}{dt} = Input - Output$$

= Rainfall + Snow + Runoff + Inflows
 - Evaporation – Withdrawal – Overflow - Seepage

AED CODE STRUCTURE



libaed-api

- aed_api
- aed_ptm
- aed_zones

libaed-water

- aed_oxygen
- aed_phosphorus
- aed_nitrogen
- aed_silica
- aed_organic_matter
- aed_phytoplankton
- aed_zooplankton
- aed_carbon
- aed_methane
- aed_ebullition
- aed_bioparticles
- aed_geochemistry
- aed_phreeqcrm
- aed_habitat_water
- aed_noncohesive
- aed_pathogens
- aed_pesticides
- aed_tracer

libaed-benthic

- aed_habitat_ benthic
- aed_macrophyte
- aed_macroalgae
- aed_bivalve

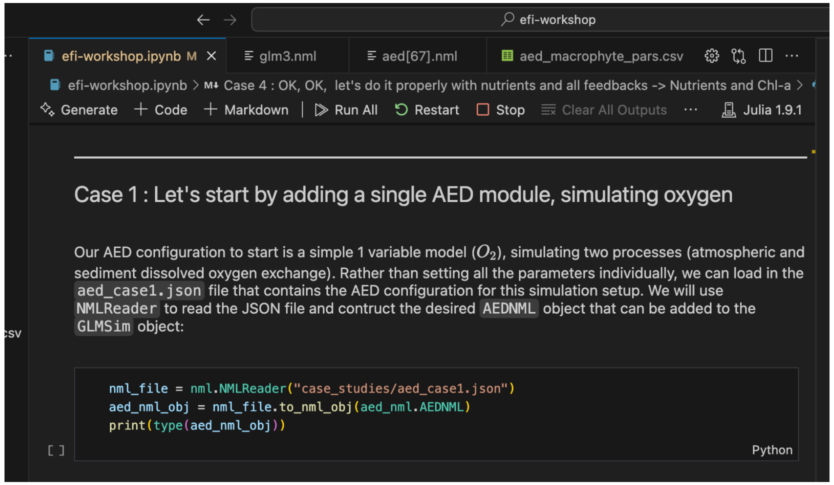
libaed-riparian

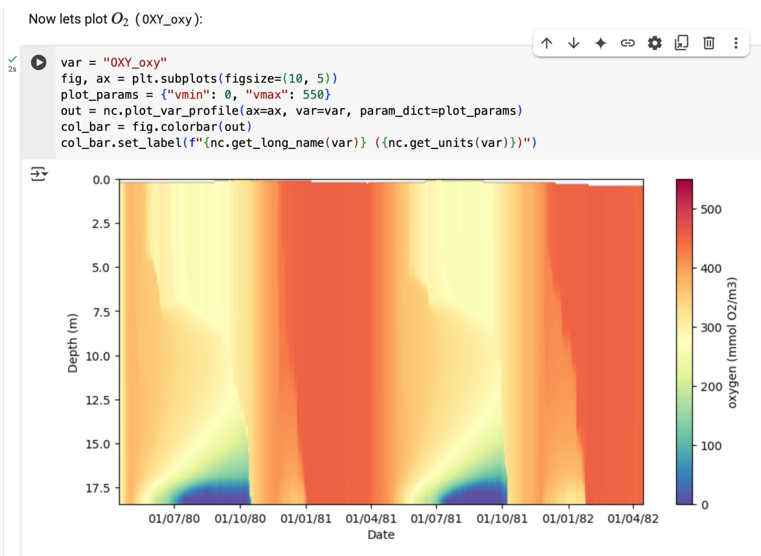
- aed_habitat_riparian
- aed_vegetation
- aed_soil
- aed_ass

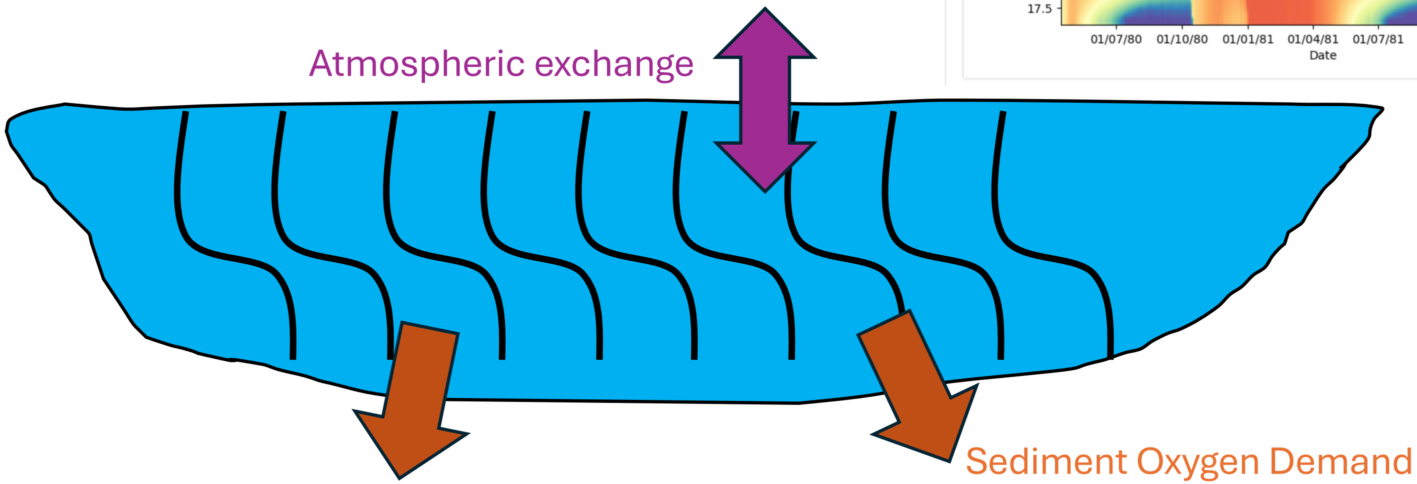
libaed-dev

- aed_oasim
- aed_seddiagenesis
- aed_sedcandi
- aed_perfect_beast
- aed_sfm

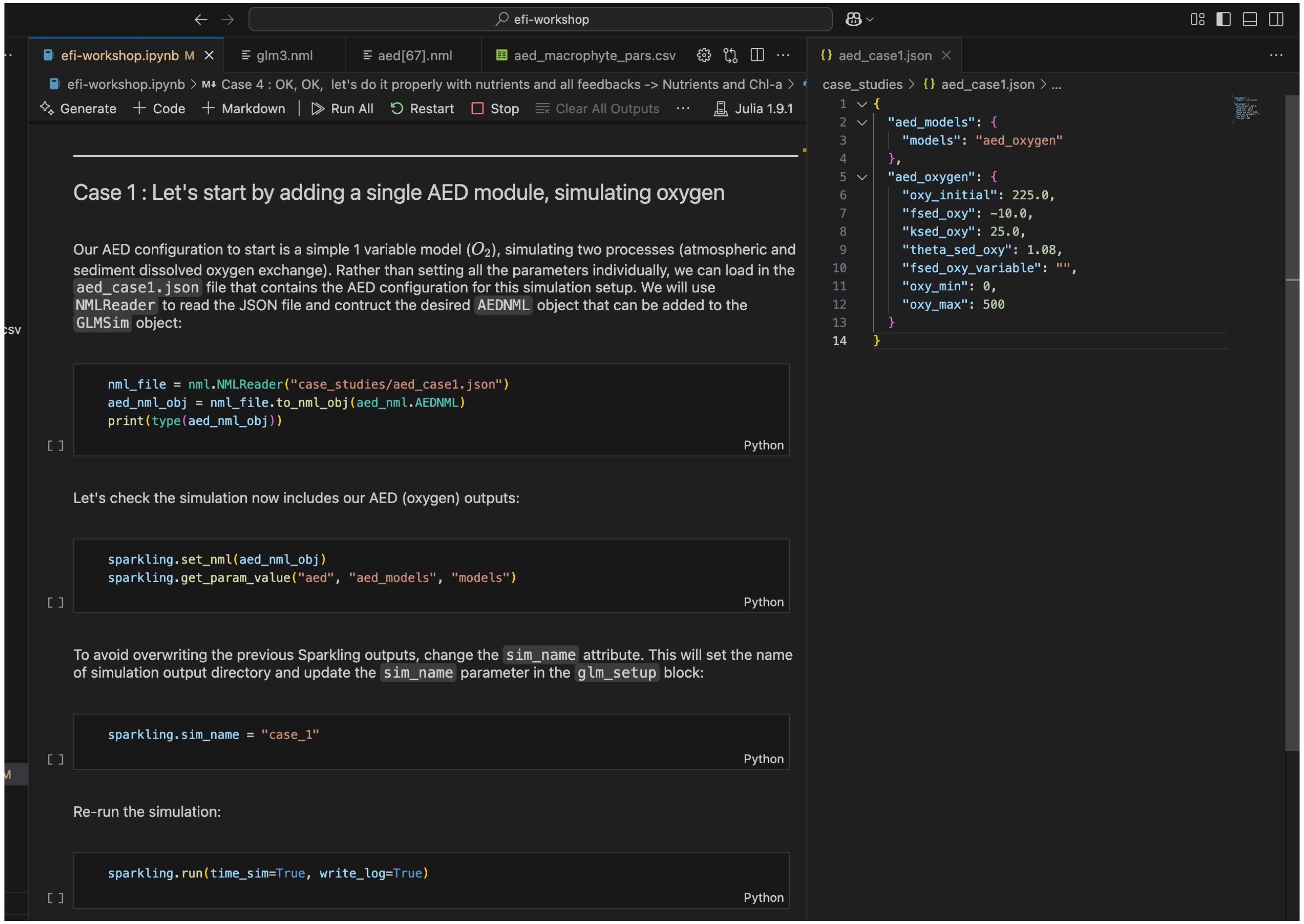
CASE 1: ADD FIRST MODULE - OXYGEN







PY BOOK: Reading JSON with AED settings, and prepares sim



efi-workshop

efi-workshop.ipynb M | glm3.nml | aed[67].nml | aed_macrophyte_pars.csv | aed_case1.json

efi-workshop.ipynb > M↓ Case 4 : OK, OK, let's do it properly with nutrients and all feedbacks -> Nutrients and Chl-a >

Generate + Code + Markdown | Run All Restart Stop Clear All Outputs ··· Julia 1.9.1

Case 1 : Let's start by adding a single AED module, simulating oxygen

Our AED configuration to start is a simple 1 variable model (O_2), simulating two processes (atmospheric and sediment dissolved oxygen exchange). Rather than setting all the parameters individually, we can load in the `aed_case1.json` file that contains the AED configuration for this simulation setup. We will use `NMLReader` to read the JSON file and contruct the desired `AEDNML` object that can be added to the `GLMSim` object:

```python
nml_file = nml.NMLReader("case_studies/aed_case1.json")
aed_nml_obj = nml_file.to_nml_obj(aed_nml.AEDNML)
print(type(aed_nml_obj))
```

Let's check the simulation now includes our AED (oxygen) outputs:

```python
sparkling.set_nml(aed_nml_obj)
sparkling.get_param_value("aed", "aed_models", "models")
```

To avoid overwriting the previous Sparkling outputs, change the `sim_name` attribute. This will set the name of simulation output directory and update the `sim_name` parameter in the `glm_setup` block:

```python
sparkling.sim_name = "case_1"
```

Re-run the simulation:

```python
sparkling.run(time_sim=True, write_log=True)
```

case_studies > {} aed_case1.json > ...

```json
{
  "aed_models": {
    "models": "aed_oxygen"
  },
  "aed_oxygen": {
    "oxy_initial": 225.0,
    "fsed_oxy": -10.0,
    "ksed_oxy": 25.0,
    "theta_sed_oxy": 1.08,
    "fsed_oxy_variable": "",
    "oxy_min": 0,
    "oxy_max": 500
  }
}
```

Inspecting manual to see processes

4.3 Model Description

This module supports one state variable to capture the oxygen concentration, O_2. The module is a low-level module that supports the two core processes of air-water exchange, $\check{f}^{O_2}_{atm}$, and sediment-water exchange, $\hat{f}^{O_2}_{sed}$, and is designed to be linked to by other modules that interact with oxygen. The dynamics of O_2 can therefore be summarised as:

$$
\begin{aligned}
\frac{D}{Dt}O_2 = \mathbb{M} + \mathcal{S} \quad & + \overbrace{\check{f}^{O_2}_{atm} + \hat{f}^{O_2}_{sed}}^{\text{aed\_oxygen}} \\
& - f^{DOC}_{min} - f^{NH_4}_{nitrf} - f^{CH_4}_{ch4ox} - f^{H_2S}_{h2sox} - f^{FeII}_{feox} \\
& + f^{PHY}_{gpp} - f^{PHY}_{rsp} - f^{ZOO}_{rsp} + \hat{f}^{MAC}_{gpp} - \hat{f}^{MAC}_{rsp} \\
& + f^{MAG}_{gpp} - f^{MAG}_{rsp} - \hat{f}^{BIV}_{rsp}
\end{aligned}
\tag{4.1}
$$

where $\mathbb{M}$ and $\mathcal{S}$ refer to water mixing and boundary source terms, respectively, and the coloured f terms reflect the optionally configurable contributions from other modules; these include the breakdown of DOC by aerobic heterotrophic bacteria to CO_2, whereby a stoichiometrically equivalent amount of oxygen is removed, chemical oxidation reactions, such as nitrification or sulfide oxidation, photosynthetic oxygen production and respiratory consumption by phytoplankton, and also oxygen consumed and produced by any benthic biological groups.

Inspecting manual to see variable names

State variables

Table 4.1: Oxygen - state variables

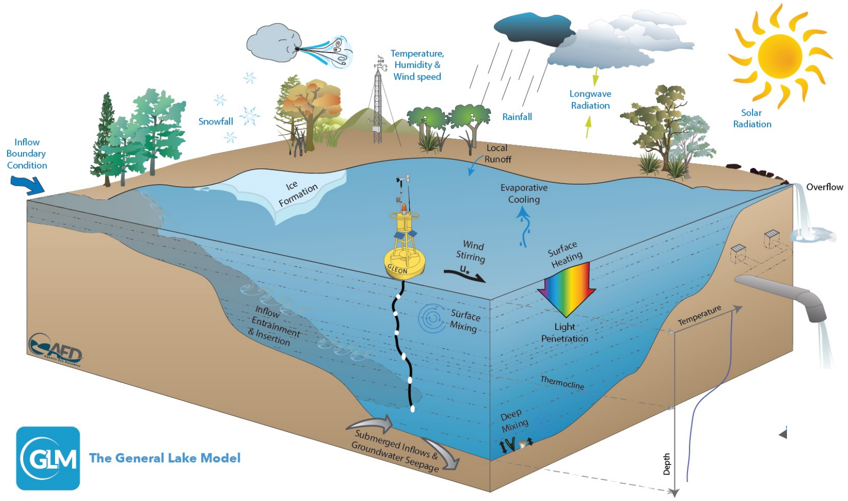
AED name	Symbol	Description	Unit	Type	Typical Range	Comments
aed_oxygen						
`OXY_oxy`	$\mathbf{O_2}$	dissolved oxygen concentration	$mmol\ O_2/m^3$	pelagic	0 - 500	.
Dependent variables						
`SDF_Fsed_oxy`	$\mathbf{F}^{oxy}_{sed}$	sediment O_2 flux	$mmol\ O_2/m^2/s$	benthic	-300 - 300	read and output as /day, but internally used as /s

Diagnostics

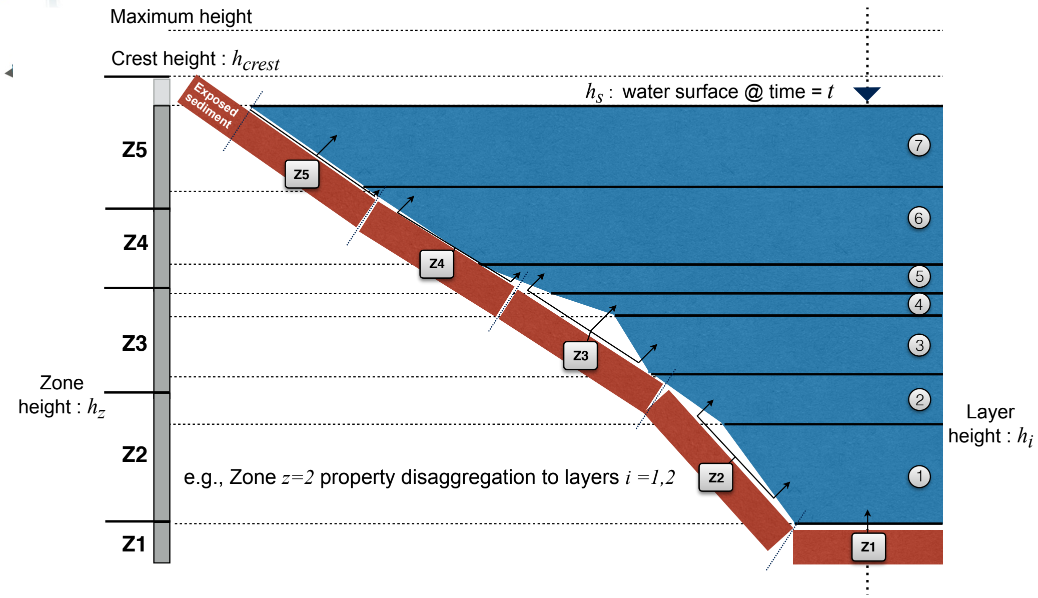
Table 4.2: Oxygen - *diagnostic* variables

AED name	Symbol	Description	Unit	Type	Typical Range	Comments
diag_level = 0+						
`OXY_sat`	$\mathbf{O_2^{sat}}$	oxygen saturation	%	pelagic	0-200	.
diag_level = 2+						
`OXY_sed_oxy`	$\mathbf{F}^{oxy}_{sed}$	O_2 exchange across sediment-water interface	$mmol\ O_2/m^2/d$	benthic	0 - 50	.
`OXY_atm_oxy_flux`	$\mathbf{F}^{oxy}_{atm}$	O_2 exchange across atm-water interface	$mmol\ O_2/m^2/d$	surface	0 - 100	.
diag_level = 9+						

CASE 2: GLM “BENTHIC ZONES”



Sediment 'zones' interact with overlaying layers for heat, O_2 and nutrient fluxes



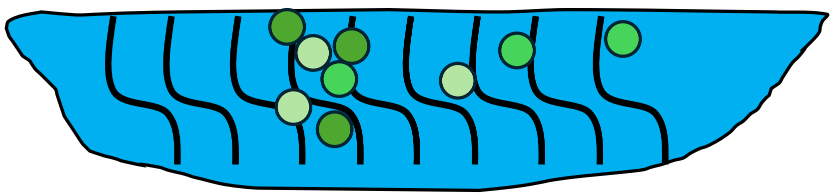
CASE 3 & 4: ADD `aed_phytoplankton`

Phytoplankton Growth Equation (10.4)

$$
f_{gpp}^{PHY_a} = [PHY_a] \times
$$

$$
\underbrace{R_{growth}^{PHY_a}}_{\text{Max growth rate at } 20^\circ\text{C}} \times \underbrace{(1 - k_{pr}^{PHY_a})}_{\text{Photorespiratory loss}} \times \underbrace{\Phi_{tem}^{PHY_a}(T)}_{\text{Temperature scaling}} \times \underbrace{\Phi_{str}^{PHY_a}(T)}_{\text{Metabolic stress}} \times
$$

$$
\min\left\{ \underbrace{\Phi_{light}^{PHY_a}(I)}_{\text{Light limitation}}, \underbrace{\Phi_{N}^{PHY_a}(NO_3, NH_4, PHY_{N_a})}_{\text{N limitation}}, \underbrace{\Phi_{P}^{PHY_a}(PO_4, PHY_{P_a})}_{\text{P limitation}}, \underbrace{\Phi_{Si}^{PHY_a}(RSi)}_{\text{Si limitation}} \right\} \times
$$



Growth limitations

```
'PHY_cyano_fI',
'PHY_cyano_fNit',
'PHY_cyano_fPho',
'PHY_cyano_fSil',
'PHY_cyano_fT',
'PHY_cyano_fSal',
```

Processes

```
'PHY_cyano_gpp_c',
'PHY_cyano_rsp_c',
'PHY_cyano_exc_c',
'PHY_cyano_mor_c',
'PHY_cyano_set_c',
'PHY_cyano_gpp_n',
'PHY_cyano_rsp_n',
'PHY_cyano_exc_n',
'PHY_cyano_mor_n',
'PHY_cyano_set_n',
'PHY_cyano_gpp_p',
'PHY_cyano_rsp_p',
'PHY_cyano_exc_p',
'PHY_cyano_mor_p',
'PHY_cyano_set_p',
```

CASE 3 & 4: `aed_phytoplankton`

- Advanced options in AED to consider when configuring a phytoplankton simulation:

  - <u>Temperature</u> optimum and maximum
  - <u>Light limitation</u> : photo-inhibited or non-photoinhibited
  - <u>Nutrient stoichiometry</u> : fixed C:N:P or luxury internal storage
  - <u>Settling or migration</u> : Constant, Stoke’s , Internal Buoyancy, Motility
  - <u>Microphytobenthos</u> (MPB) pool

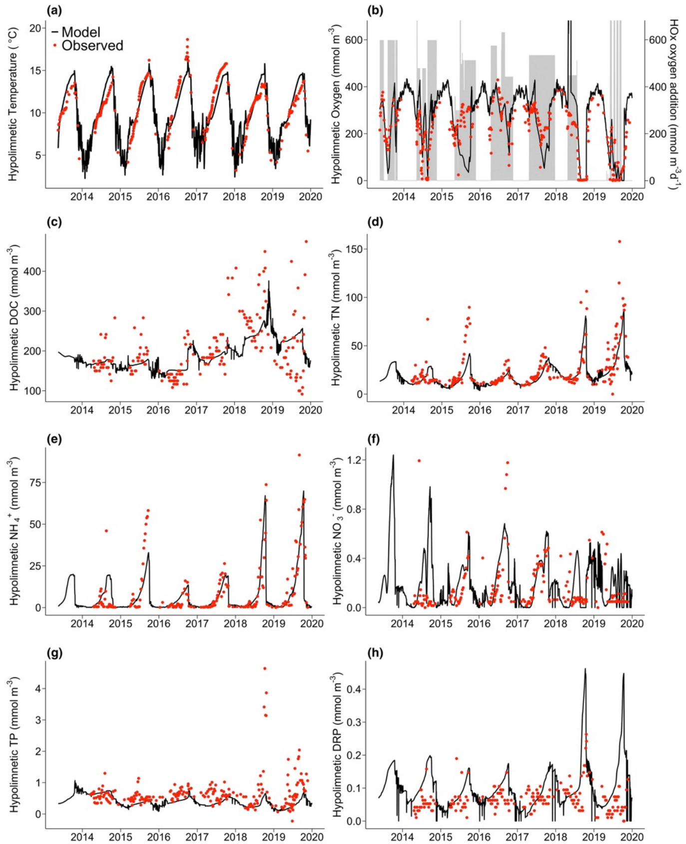
CASE 4 & 5: NUTRIENT / PHYTOPLANKTON APPLICATION

GLM 1D: 2013-2020

- *Met*
- *Inflow*
- *Outflows*
- *Side-stream oxygenation*

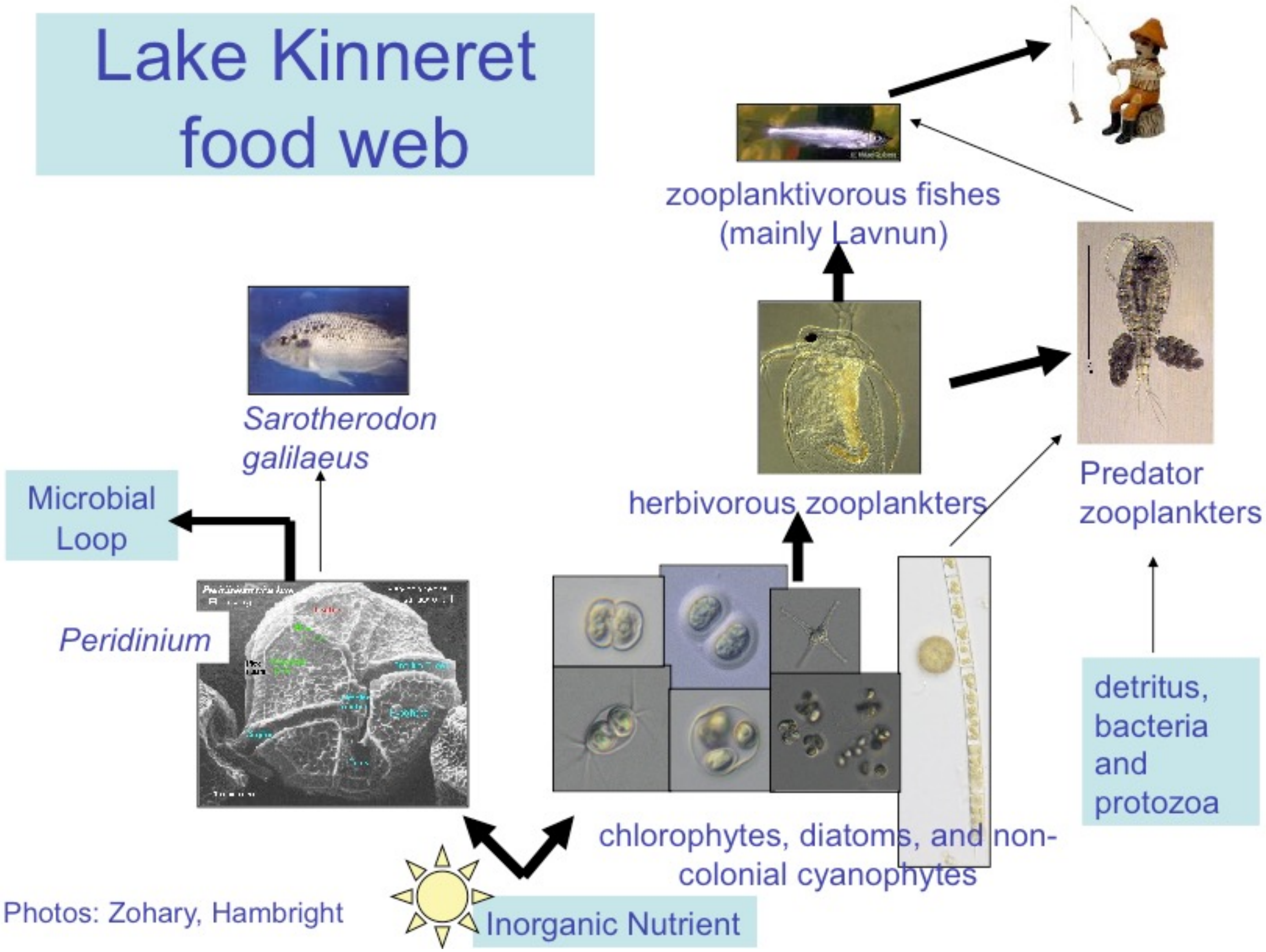
AED (Aquatic EcoDynamics) model applied to simulate:

- *DO*
- *DOM*
- *POM*
- *NH_4, NO_3*
- *PO_4*



Carey, C.C., Hanson, P.C., Thomas, R.Q., Gerling, A.B., Hounshell, A.G., Lewis, A.S., Lofton, M.E., McClure, R.P., Wander, H.L., Woelmer, W.M. and Niederlehner, B.R., 2022. **Anoxia decreases the magnitude of the carbon, nitrogen, and phosphorus sink in freshwaters**. Global Change Biology, 28(16), pp.4861-4881.

CASE 6: ADD `aed_zooplankton`



SEQUENTIONAL VS DISTRIBUTED SIMULATION

sequential_sim.py U

sequential_sim.py > ...

```python
import random

from glmpy import example_sims


sparkling = example_sims.SparklingSim(outputs_dir="sequential_sims")
num_sims = 50
kw = [random.uniform(0, 1) for i in range(0, num_sims)]

for i in range(0, num_sims):
    sparkling.sim_name = f"sparkling_{i}"
    sparkling.glm_nml.blocks["light"].params["Kw"].value = kw[i]
    sparkling.run()
```

bash

```
root@81568ab23dc4:/workspaces/glm-py# python sequential_sim.py
```

multi_sim.py U

multi_sim.py > ...

```python
import random

from glmpy import example_sims, sim


sparkling = example_sims.SparklingSim(outputs_dir="multi_sims")
num_sims = 50
kw = [random.uniform(0, 1) for i in range(0, num_sims)]

glm_sims = []
for i in range(0, num_sims):
    glm_sim = sparkling.get_deepcopy()
    glm_sim.sim_name = f"sparkling_{i}"
    glm_sim.glm_nml.blocks["light"].params["Kw"].value = kw[i]
    glm_sims.append(glm_sim)

multi_sim = sim.MultiSim(glm_sims)
multi_sim.run()
```

bash

```
root@81568ab23dc4:/workspaces/glm-py# python multi_sim.py
```

FEEDBACK & FURTHER INFORMATION

- EFI AED clinics : Thursday 9 – 11 AM -> Patio
- CONTACTS : matt.hipsey@uwa.edu.au
  giles.knight@uwa.edu.au
- SUGGESTIONS & FEATURE REQUESTS:
  https://github.com/AquaticEcoDynamics/efi-workshop/issues