



# Introduction to process-based lake modeling in R using GLM (General Lake Model)

Experiment on your lakes safely from home on your laptop

**Robert Ladwig, A. Adhlakha, H.A. Dugan & P.C. Hanson**

University of Wisconsin-Madison



rladwig2@wisc.edu



@hydrobert



supported by NSF ABI development grant, #DBI 1759865

# Who's who?

Robert, postdoc at  
Center for Limnology



Hilary, Assistant Prof.  
UW-Madison



Aryan, undergrad in  
Computer Science



Paul, Research Prof.  
UW-Madison

# Welcome!

- If you want to run the simulations during the workshop, you will need to install the following software on your computer. If you just want to watch, ask questions, and drive from the back seat, that's fine, too!
- Future questions? Email Robert at [rladwig2@wisc.edu](mailto:rladwig2@wisc.edu)

## Two paths to do the workshop examples:

- (1) Use the docker: <https://hub.docker.com/r/hydrobert/glm-workshop> (requires docker)
  - (a) this includes Rocker, all packages, all scripts and all data; then open a terminal

```
docker pull hydrobert/glm-workshop
```

```
docker run -rm -d -p 8000:8000 -e ROOT=TRUE -e PASSWORD=password hydrobert/glm-workshop:latest
```

```
open any web browser and type 'localhost:8000' (user: rstudio, password: password)
```

- (2) Clone or download files from: [https://github.com/robertladwig/GLM\\_workshop](https://github.com/robertladwig/GLM_workshop)
  - (a) you'll need R (>= 3.5) and these packages: GLM3r, glmtools, rLakeAnalyzer, tidyverse

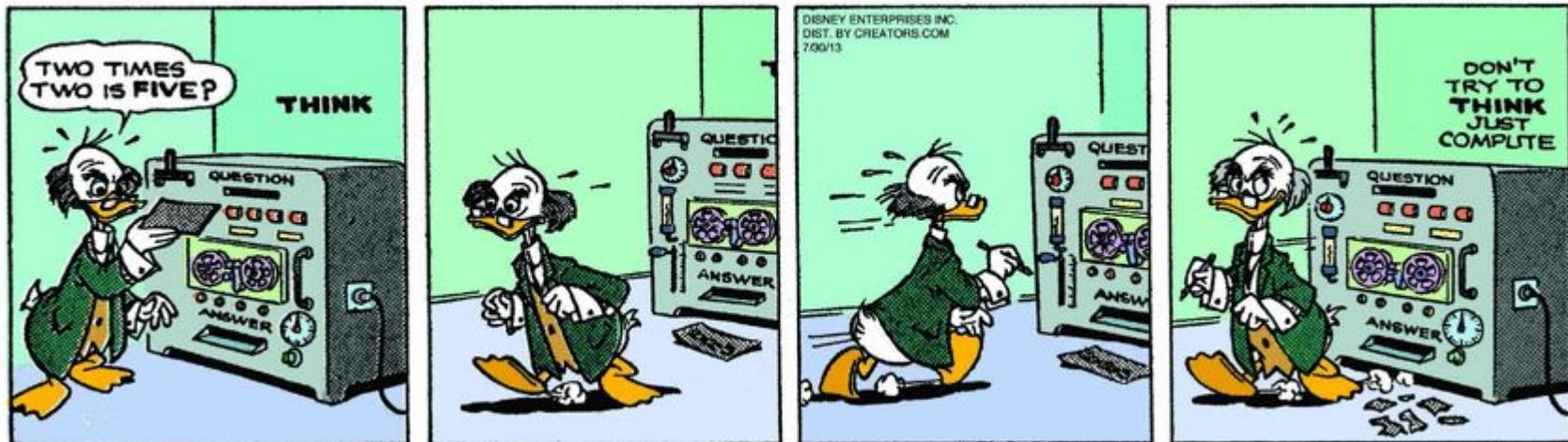


# Expectations for today

- Introduction to process-based modeling intended for all skill levels and background
- Overview of concepts and lake modeling theory → **General Lake Model (GLM)**
- **Coding examples** in R using data from Lake Mendota (thanks to NTL-LTER!)



Great textbook: Steven C. Chapra (2008) "Surface Water-Quality Modeling" Waveland Pr Inc



# Time schedule today

12:00-1:00

## Introduction to process-based lake modeling

- What is process-based modeling?
- GLM theory and applications
- AED2 theory and examples for O<sub>2</sub> and C
- Overview of R-packages

1:10-2:00

## Using the model in R

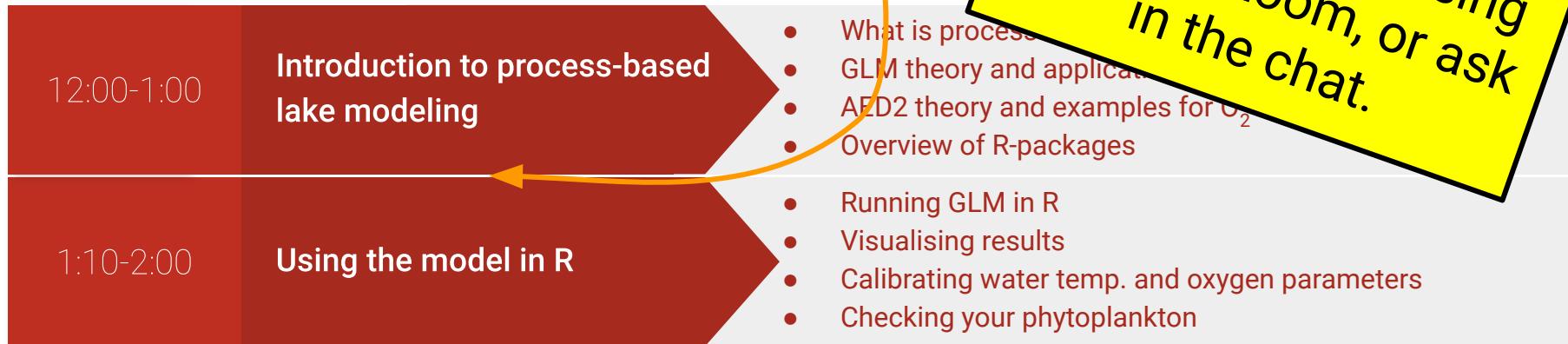
- Running GLM in R
- Visualising results
- Calibrating water temp. and oxygen parameters
- Checking your phytoplankton

# Time schedule today

12:00-1:00	Introduction to process-based lake modeling	<ul style="list-style-type: none"><li>• What is process-based modeling?</li><li>• GLM theory and application</li><li>• AED2 theory and examples for <math>O_2</math></li><li>• Overview of R-packages</li></ul>
1:10-2:00	Using the model in R	<ul style="list-style-type: none"><li>• Running GLM in R</li><li>• Visualising results</li><li>• Calibrating water temp. and oxygen parameters</li><li>• Checking your phytoplankton</li></ul>

Please ask urgent questions by "raising hand" in Zoom, or ask in the chat.

# Time schedule today



# Modeling introduction

# Process-based modeling in a nutshell

- Models are simplified reflections of reality
- Deterministic (events at next time step depend on events from previous time step)
  - and describe processes with mathematical equations
  - that are either based on empirical knowledge or physical principles
- Numerical models need time and space discretization
- As well as initial data (to start from) and boundary data (as driving data)

State of the environment



$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = q_0$$

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \nabla \mathbf{v} = -\frac{1}{\rho} \nabla p + g + \nu \Delta \mathbf{v}$$



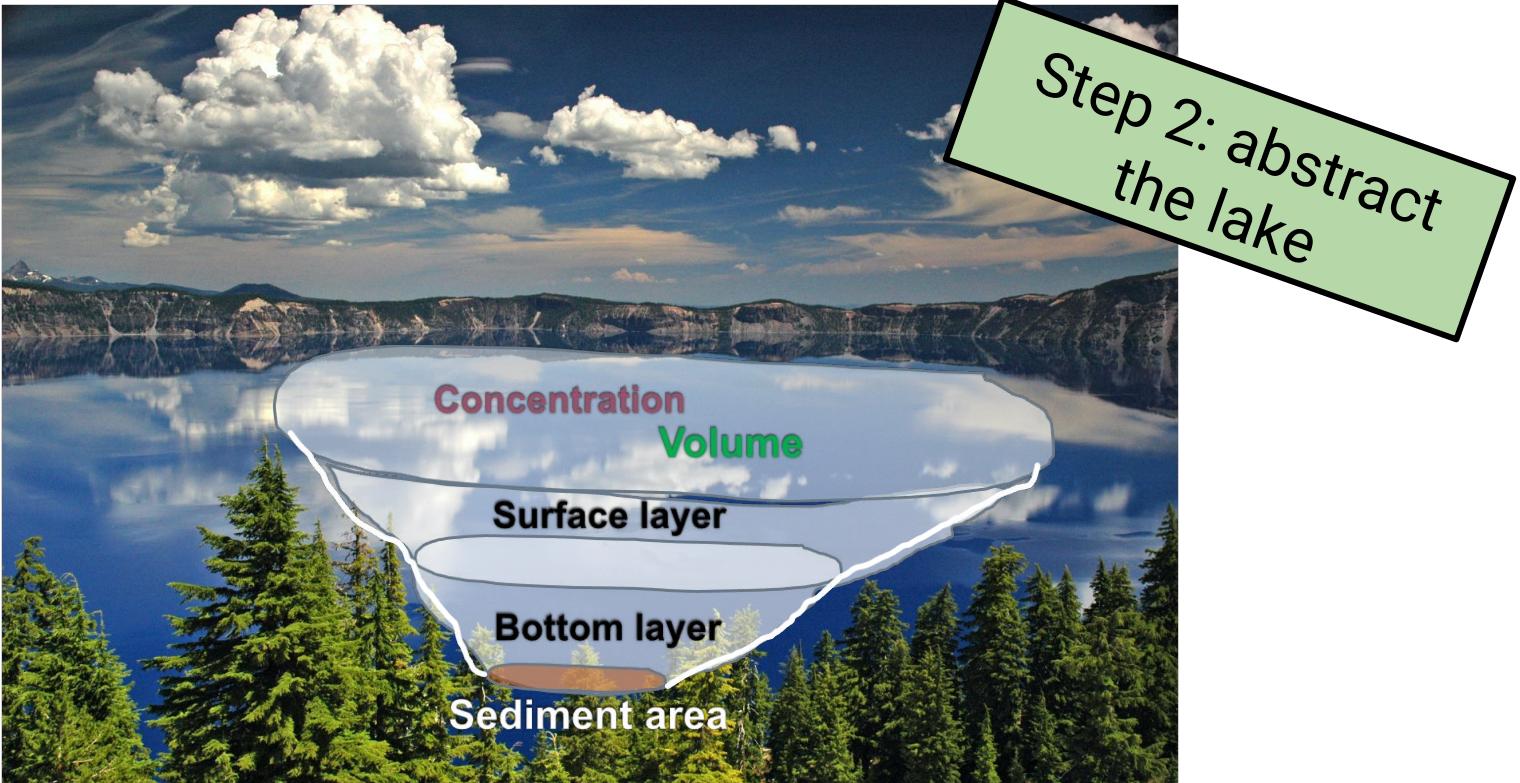
State of the system

# Process-based modeling in a nutshell

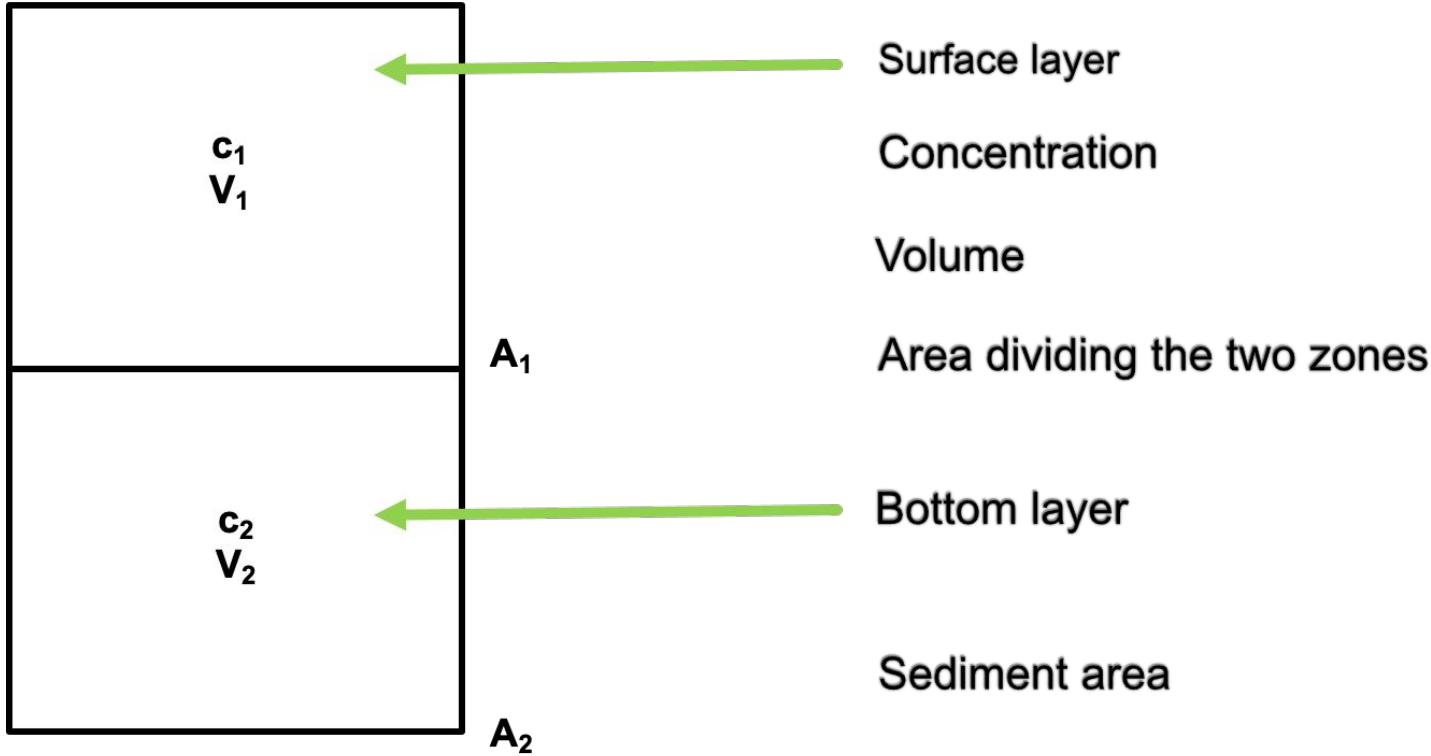


*Step 1: imagine a  
(nice) lake*

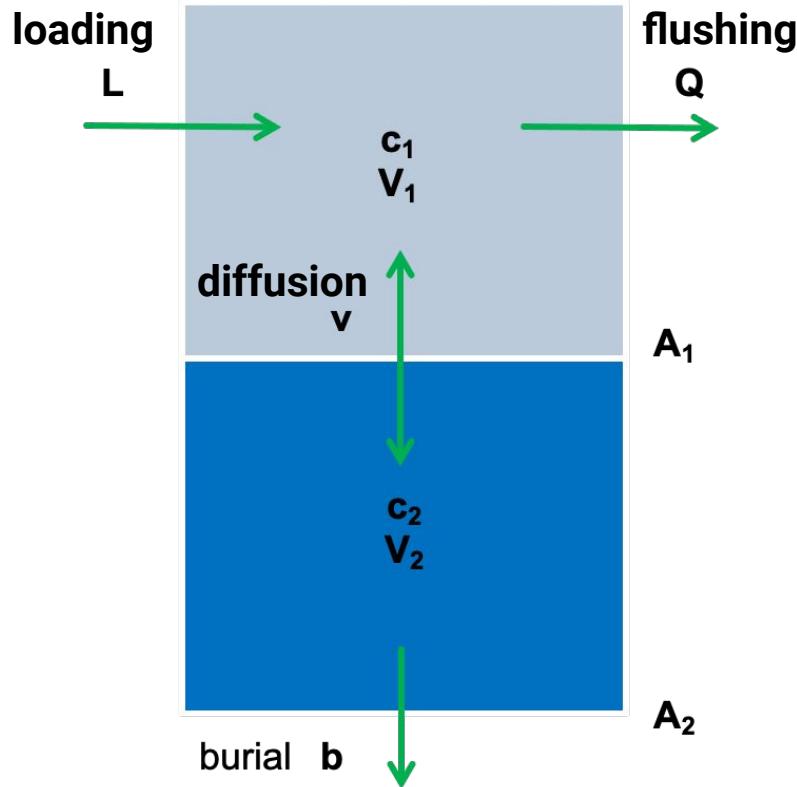
# Process-based modeling in a nutshell



# Process-based modeling in a nutshell



# Process-based modeling in a nutshell

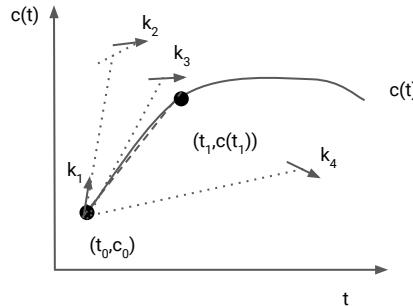
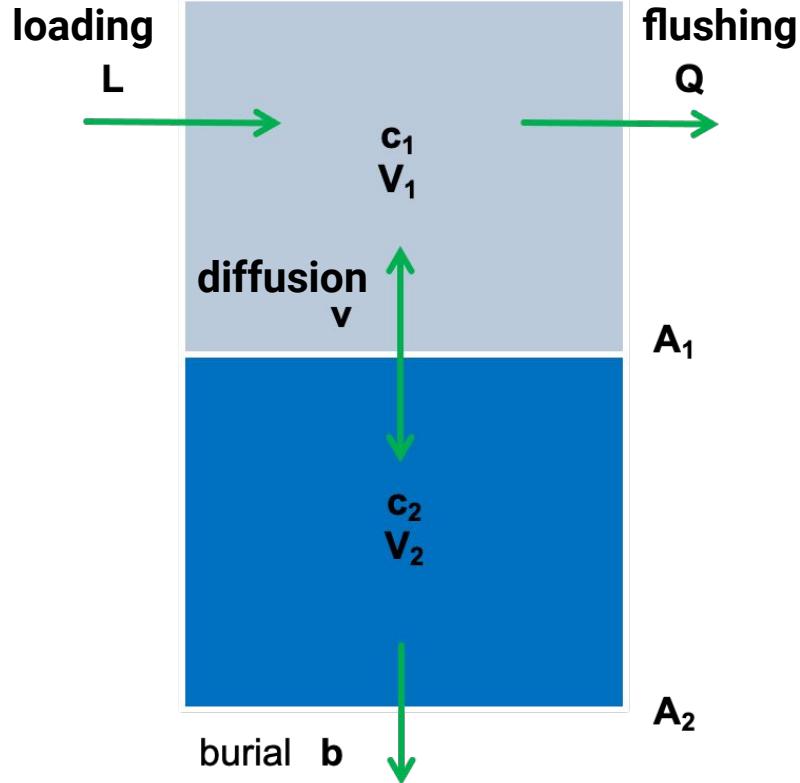


Step 3: add math

$$\frac{dc_1}{dt} = \frac{L - Qc_1 + vA_1(c_2 - c_1)}{V_1}$$

$$\frac{dc_2}{dt} = \frac{vA_1(c_1 - c_2) - bA_2c_2}{V_2}$$

# Process-based modeling in a nutshell



*Step 4: solve with numerics, e.g., Runge-Kutta*

$$c_{i+1} = c_i + f(c_i, t_i) * \Delta t \quad h = \Delta t$$

$$c_{i+1} = c_i + [\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)]\Delta t$$

$$k_1 = f(t_i, c_i)$$

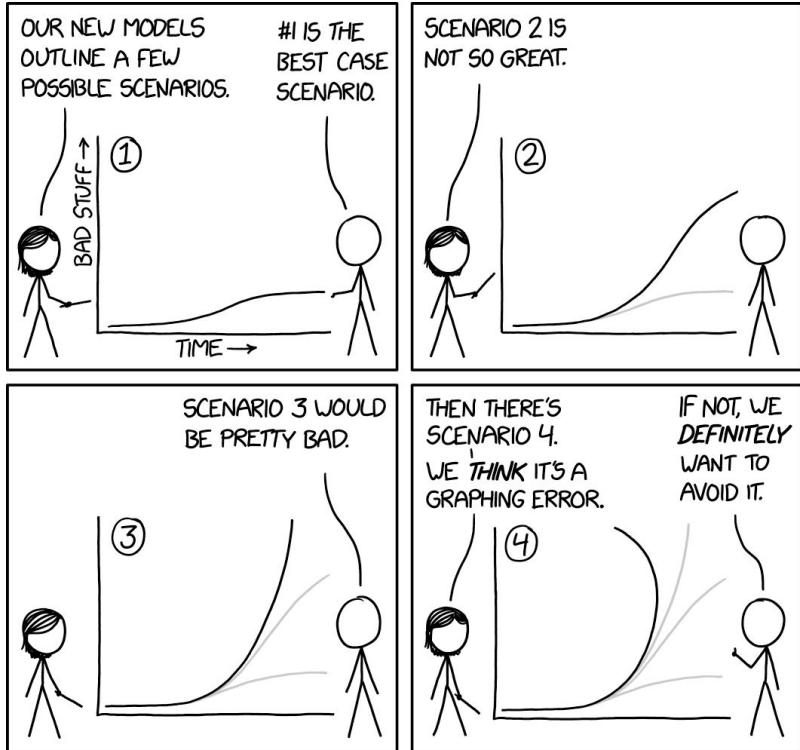
$$k_2 = f(t_i + \frac{1}{2}h, c_i + \frac{1}{2}hk_1)$$

$$k_3 = f(t_i + \frac{1}{2}h, c_i + \frac{1}{2}hk_2)$$

$$k_4 = f(t_i + h, c_i + hk_3)$$

# Process-based modeling in a nutshell

- Why should you use a model?
- Explore ideas regarding ecological systems that may not be possible/feasible to field-test for
  - logistical reasons
  - political reasons
  - financial reasons
  - physical reasons
- Formulating a model helps to better understand the system and can identify data needs
- “Your model is your hypothesis.” ([Kate](#))

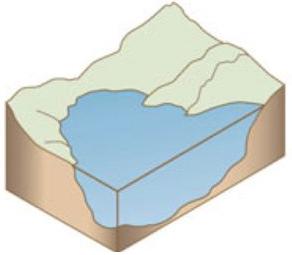


# GLM

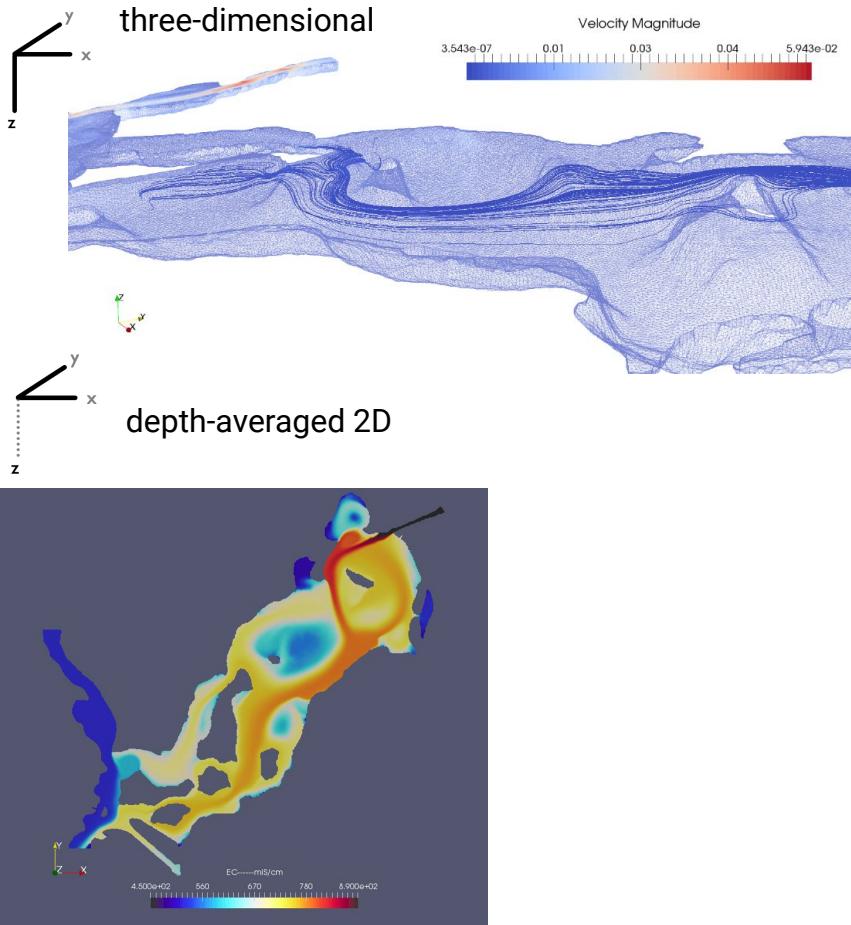
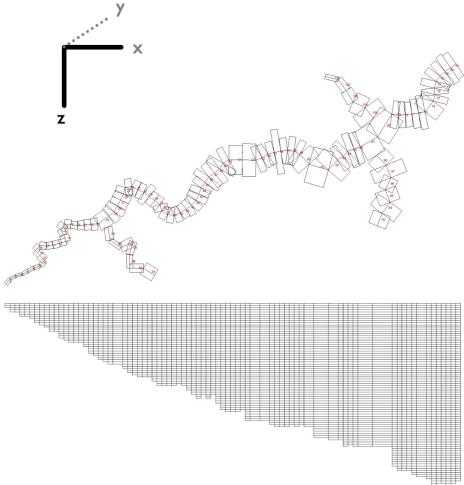
# introduction

# Multi-dimensional models

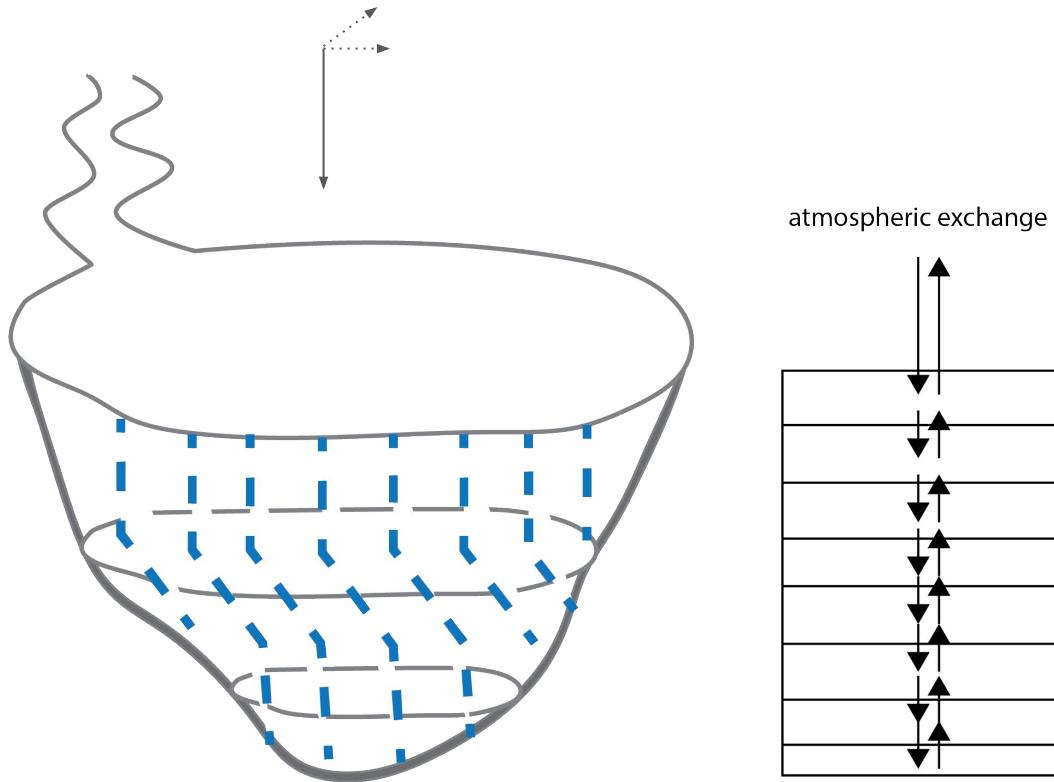
zero-dimensional  
0D: Box model



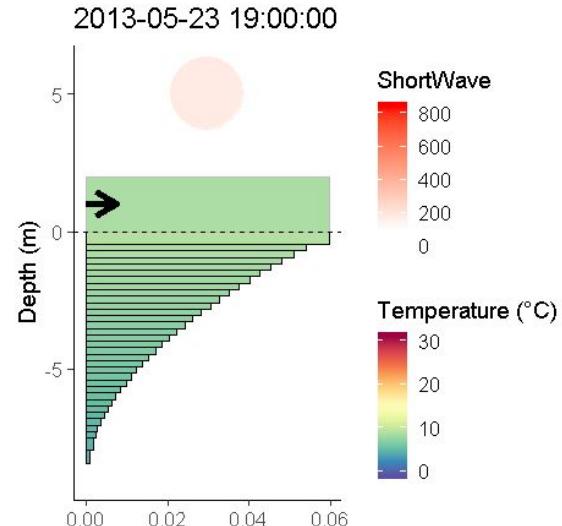
longitudinal-vertical 2D



# Process-based lake modeling



- Vertical 1D model:
  - Horizontal homogeneity
  - No Coriolis force
- Layering over vertical axis
- Every layer = volume of lake
- **Focus on heat transport**



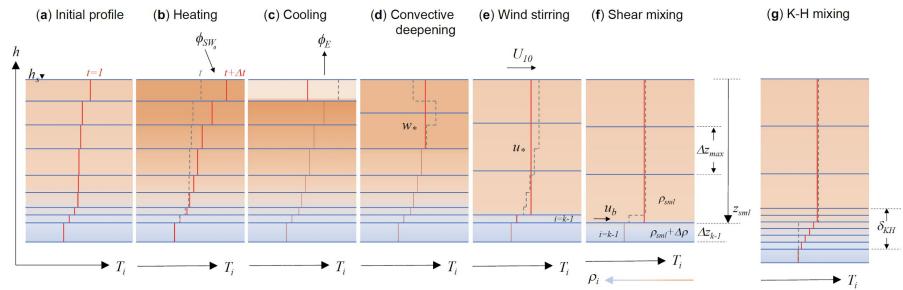
# Process-based lake modeling

- Reactive-diffusive transport equation for water temperature
- Turbulence-closure scheme to solve for momentum terms and diffusivities (e.g., k- $\epsilon$  in Simstrat)
- Molecular diffusivity is mostly neglected → approximation of  $K_z$  (e.g., MyLake)
- Alternatively, energy-balance approaches (e.g., GLM): potential ag. available kinetic energy

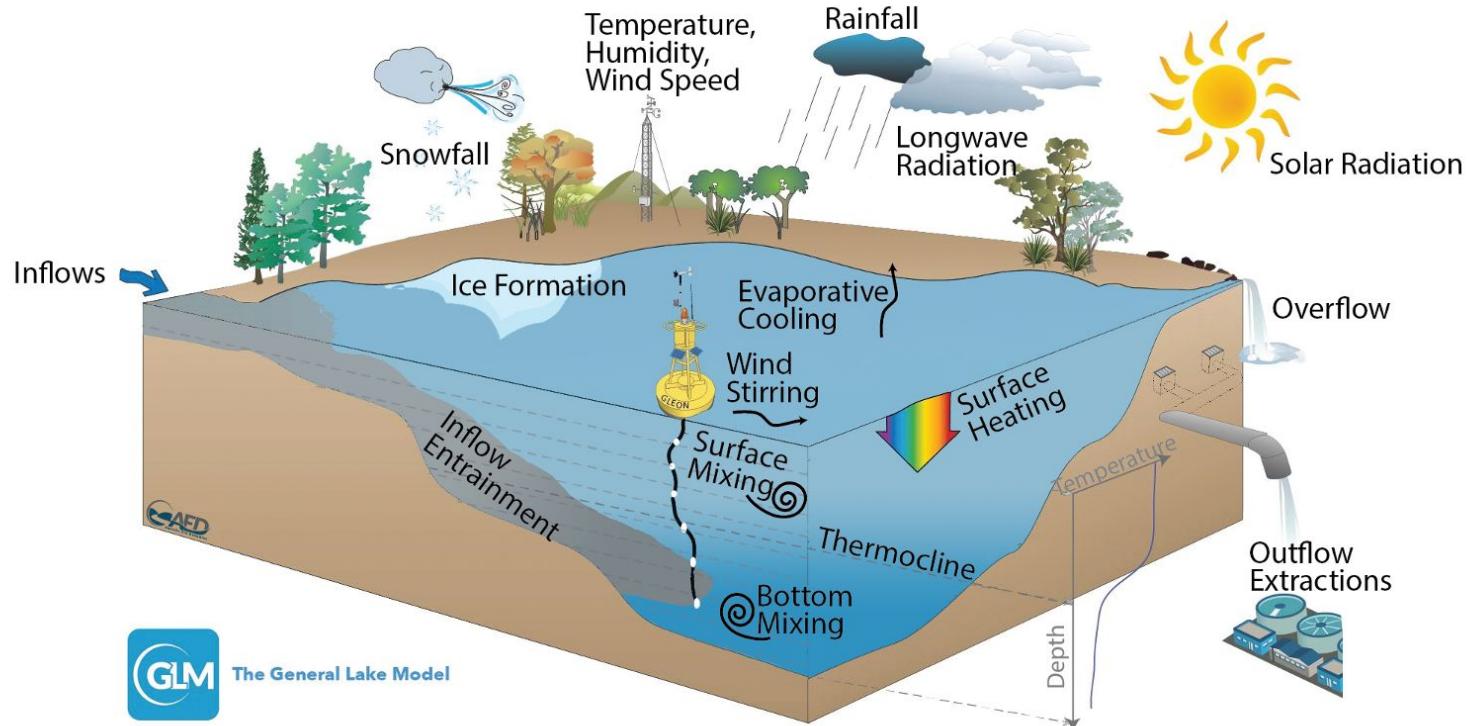
$$\frac{\partial T}{\partial t} = \frac{1}{A} \frac{\partial}{\partial z} \left( A(v_t^c + v^i) \frac{\partial T}{\partial z} \right) + \frac{1}{\rho_0 c_p} \frac{\partial H_{sol}}{\partial z} + \frac{dA}{dz} \frac{H_{geo}}{A \rho_0 c_p}$$

diffusive transport      atmospheric heating      sediment heating

$$\frac{\partial T}{\partial t} = \frac{1}{A} \frac{\partial}{\partial z} \left( A K_z \frac{\partial T}{\partial z} \right) + \frac{1}{\rho_0 c_p} \frac{\partial H_{sol}}{\partial z}$$

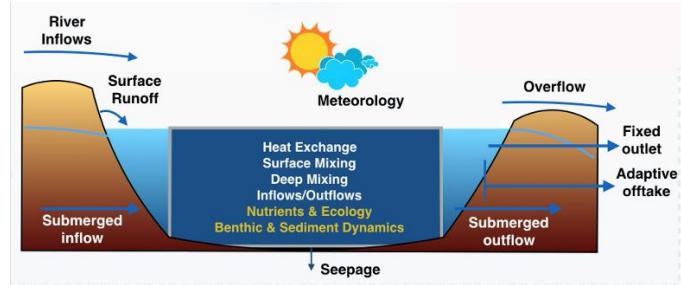


# General Lake Model (GLM)



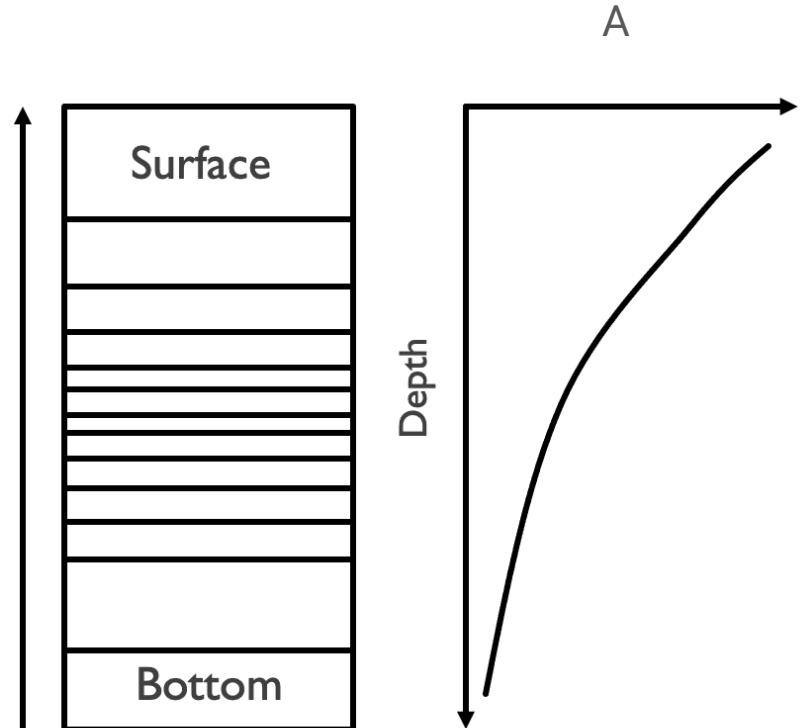
# General Lake Model (GLM)

- GLM started as a GLEON project
- Developed by Matt Hipsey, Louise Bruce and Casper Boon at UWA
- Designed to operate with the Aquatic EcoDynamics Model Libraries (AED2) → water quality model
- Written in C and Fortran
- Freely available as open-source program:  
<https://github.com/AquaticEcoDynamics/GLM>



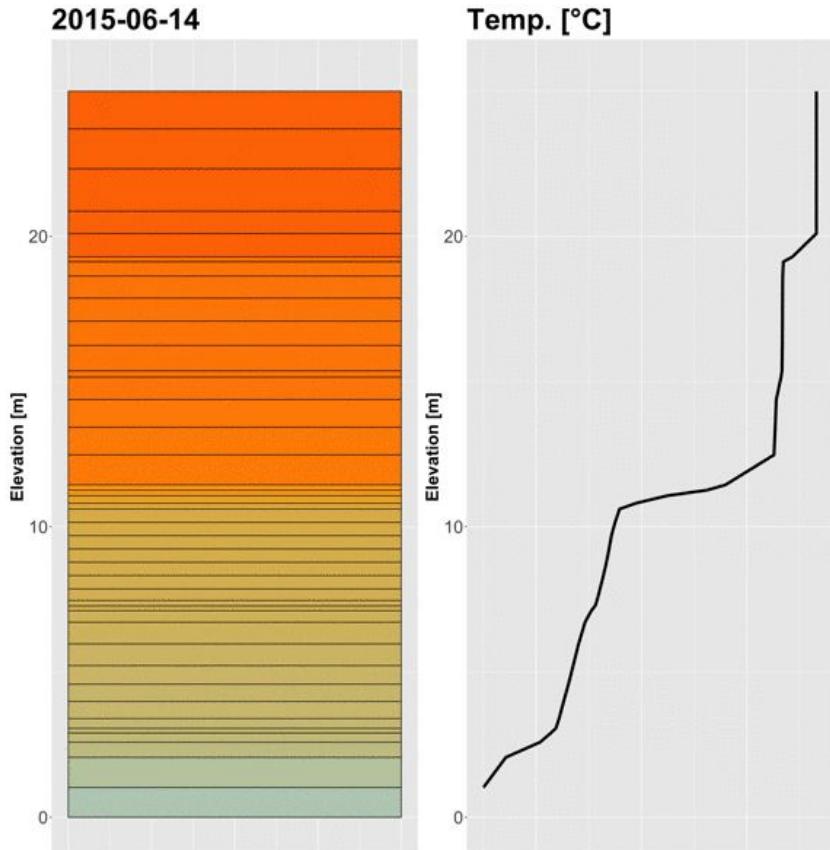
# Grid

- Each vertical layer (= control volume) adapts its thickness due to flows, mixing and mass fluxes  
→ flexible Lagrangian structure
- Each layer has 'unique' density
- When there's enough energy to overcome density gradient → layers merge
- Layer's volume changes depending on hypsography



# Grid

- Each vertical layer (= control volume) adapts its thickness due to flows, mixing and mass fluxes
  - flexible Lagrangian structure
- Each layer has 'unique' density
- When there's enough energy to overcome density gradient → layers merge
- Layer's volume changes depending on hypsography



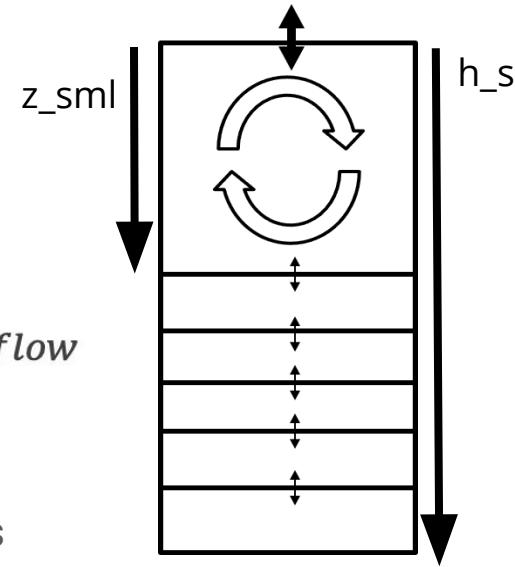
# Water balance

- Net water flux over total lake (solved by integration over time):

$$\frac{dV_s}{dt} = A_s \frac{dh_s}{dt} + \sum_I^N Q_{inf,I} - \sum_I^N Q_{out,I} - Q_{seepage} - Q_{overflow}$$

Surface layer height

- Code solves first for surface flux changes, then for all other fluxes



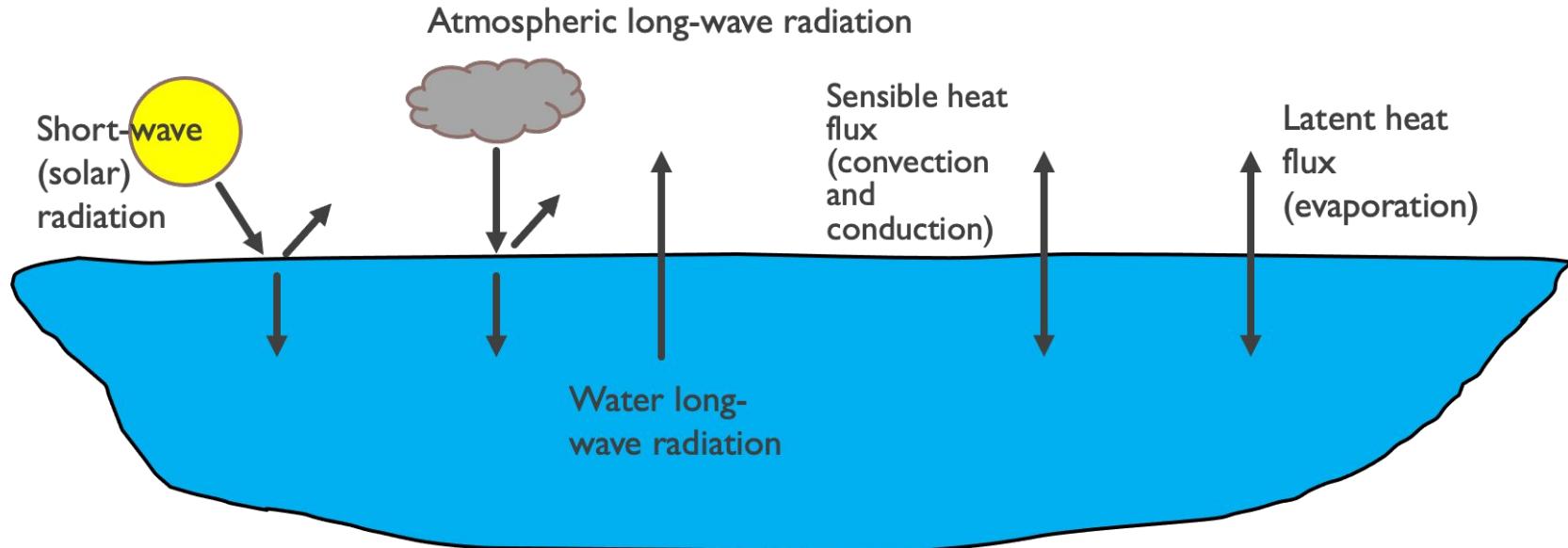
Mass balance of surface layer = precipitation + snowfall + run-off – evaporation - ice

$$\frac{dh_s}{dt} = P + S + \frac{Q_R}{A_S} - E - \frac{d\Delta z_{ice}}{dt}$$

- Mixing only affects thickness of surface layer ( $z_{sml}$ ) not its height

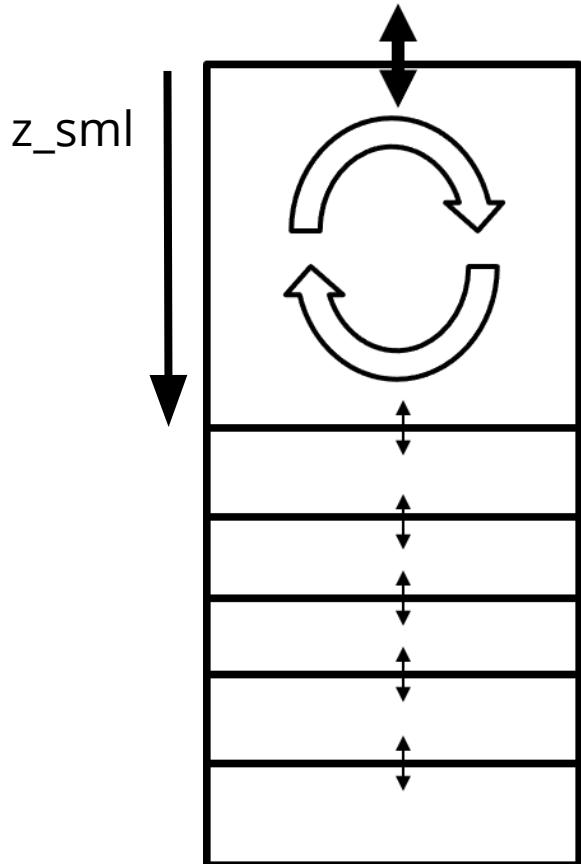
# Surface energy balance

$$\rho C_p z_s \frac{dT_s}{dt} = J_{SW} + J_{LW,in} - J_{LW,out} - J_E + J_H$$

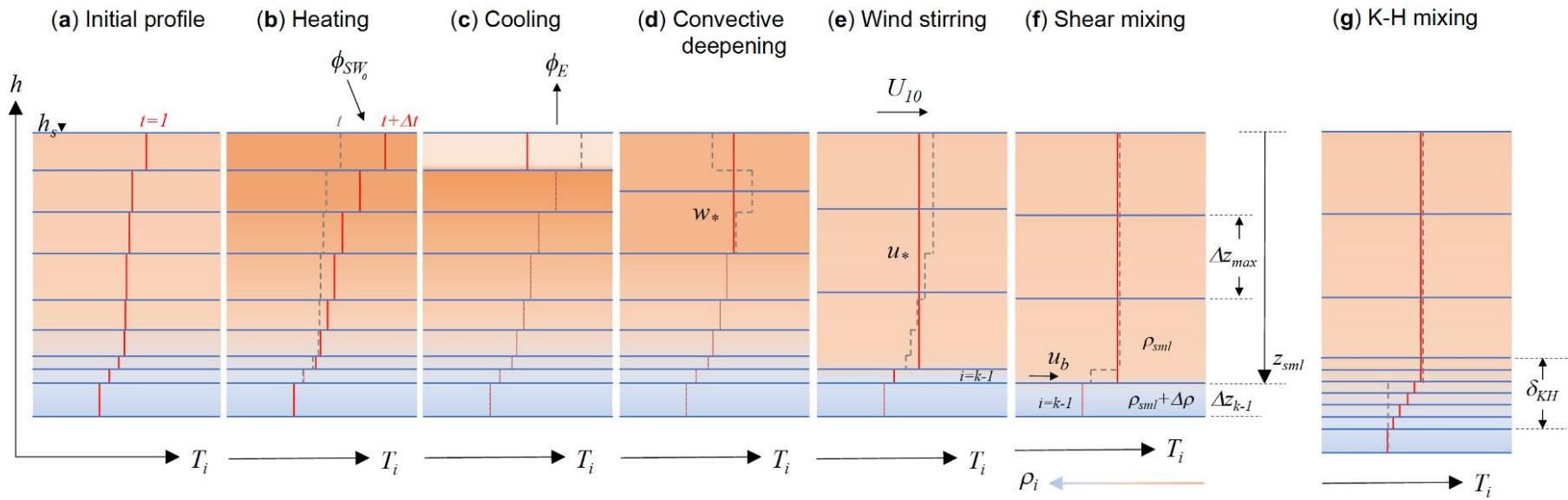


# Mixing I

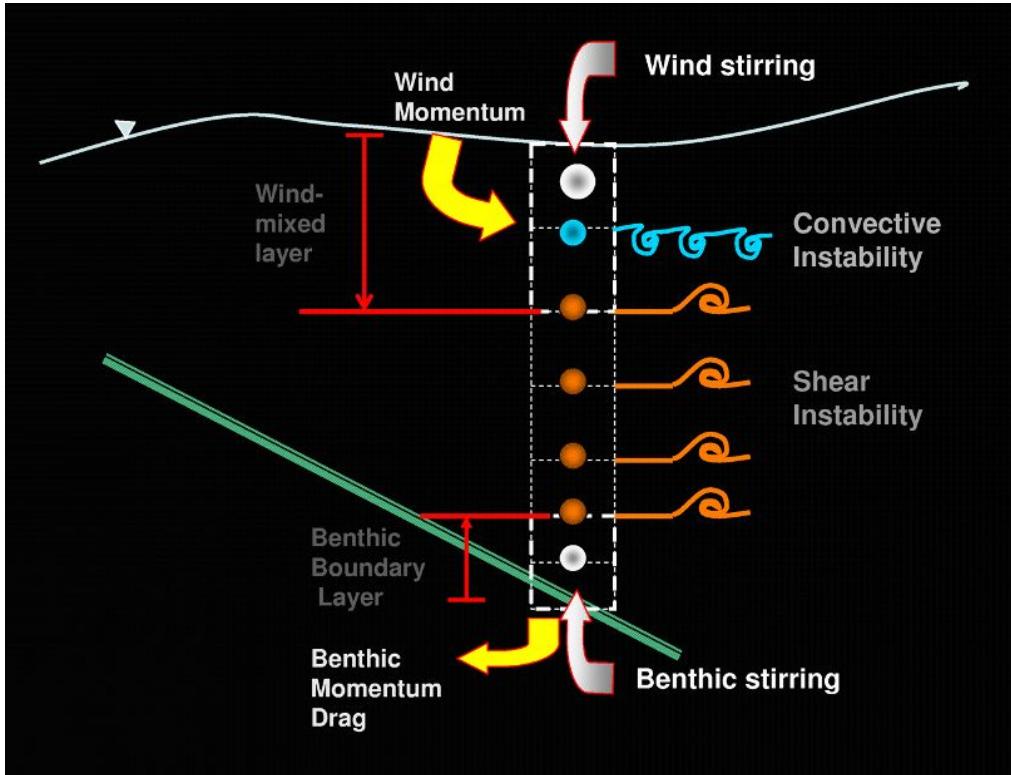
- Surface layer: energy balance approach
  - Energy balance determines thickness of surface mixed layer
  - Below surface mixed layer, turbulent diffusion approach is used
- Here, mixing depends on difference between potential energy and available kinetic energy
- Kinetic energy = convective overturn, wind stirring and shear production/Kelvin-Helmholtz billowings



# Mixing II



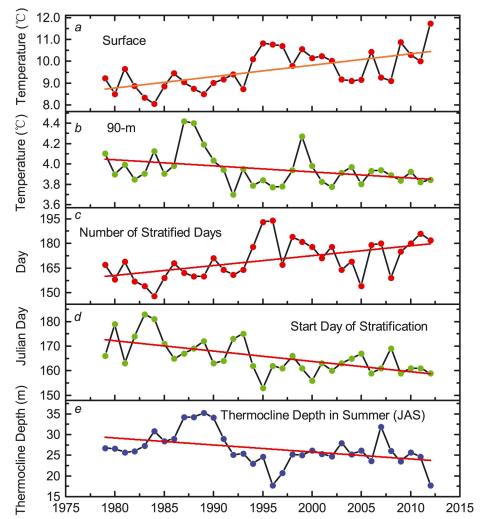
# Mixing III



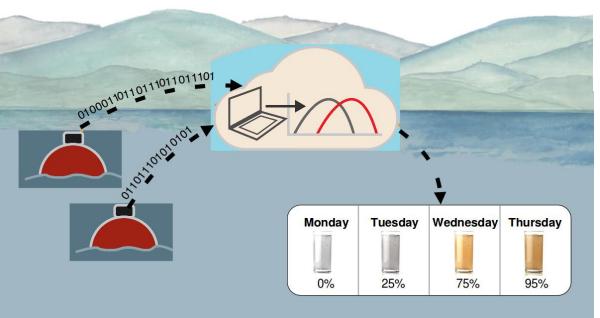
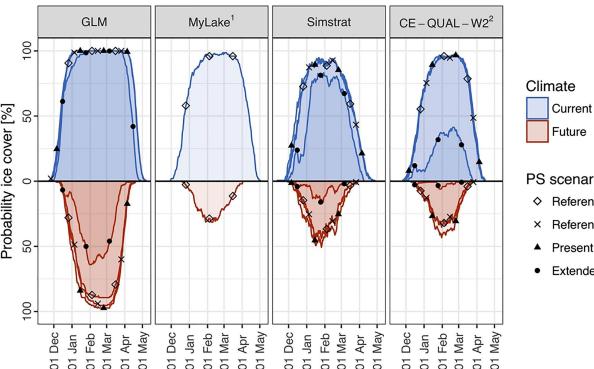
Kelvin-Helmholtz instabilities

# Example applications

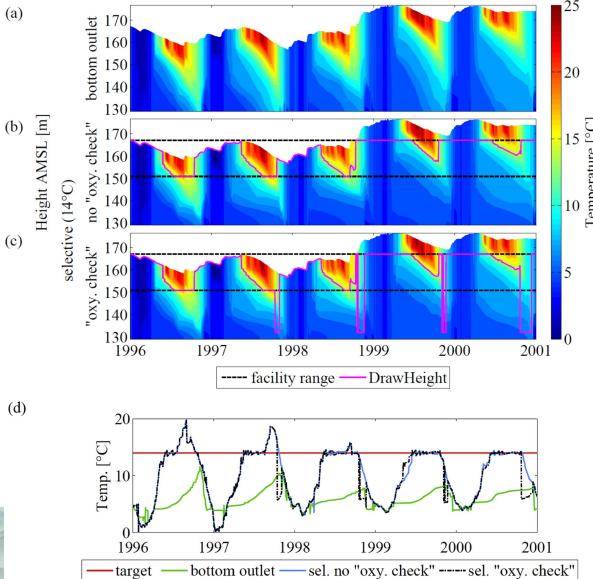
## Climate change effects on lakes<sup>1</sup>



## Future ice cover shifts<sup>2</sup>



## Optimal withdrawal depth in reservoirs<sup>4</sup>



## Smart ecological forecasting (FLARE)<sup>3</sup>

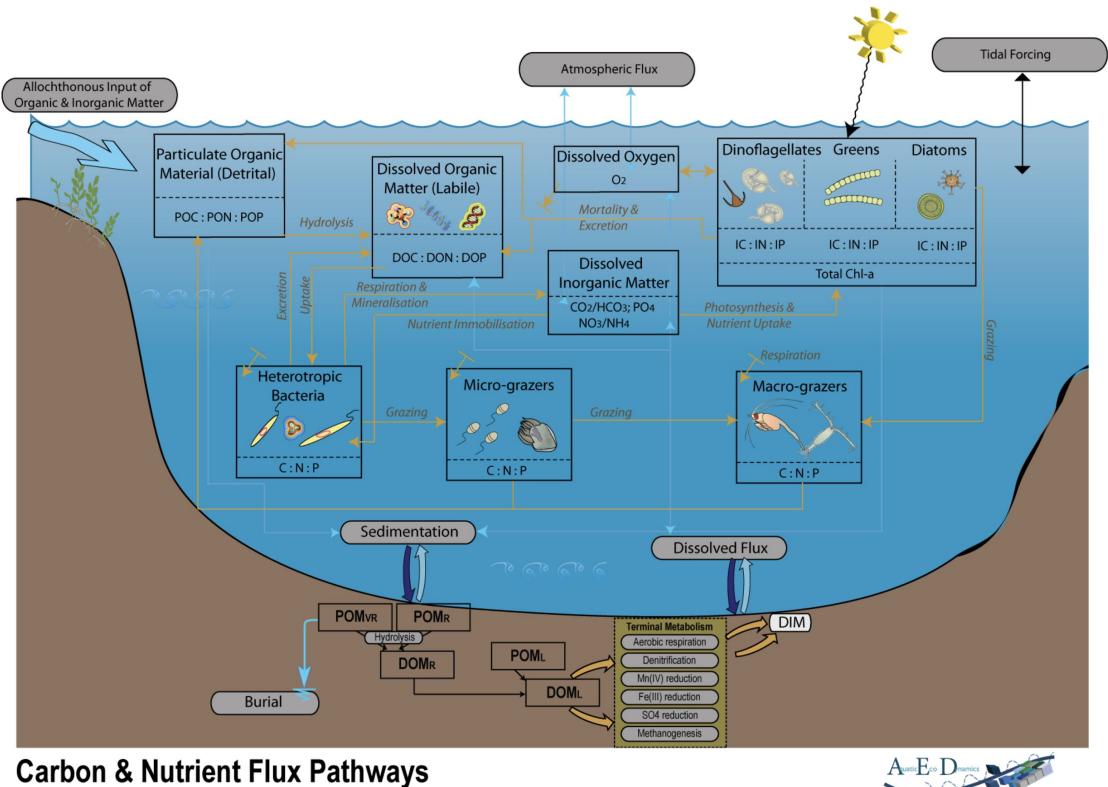
<sup>1</sup>Huang et al. 2017; <sup>2</sup>Kobler et al. 2019; <sup>3</sup>Figueiredo, Thomas, Carey & Weathers; <sup>4</sup>Weber et al. 2017

# AED2

# theory

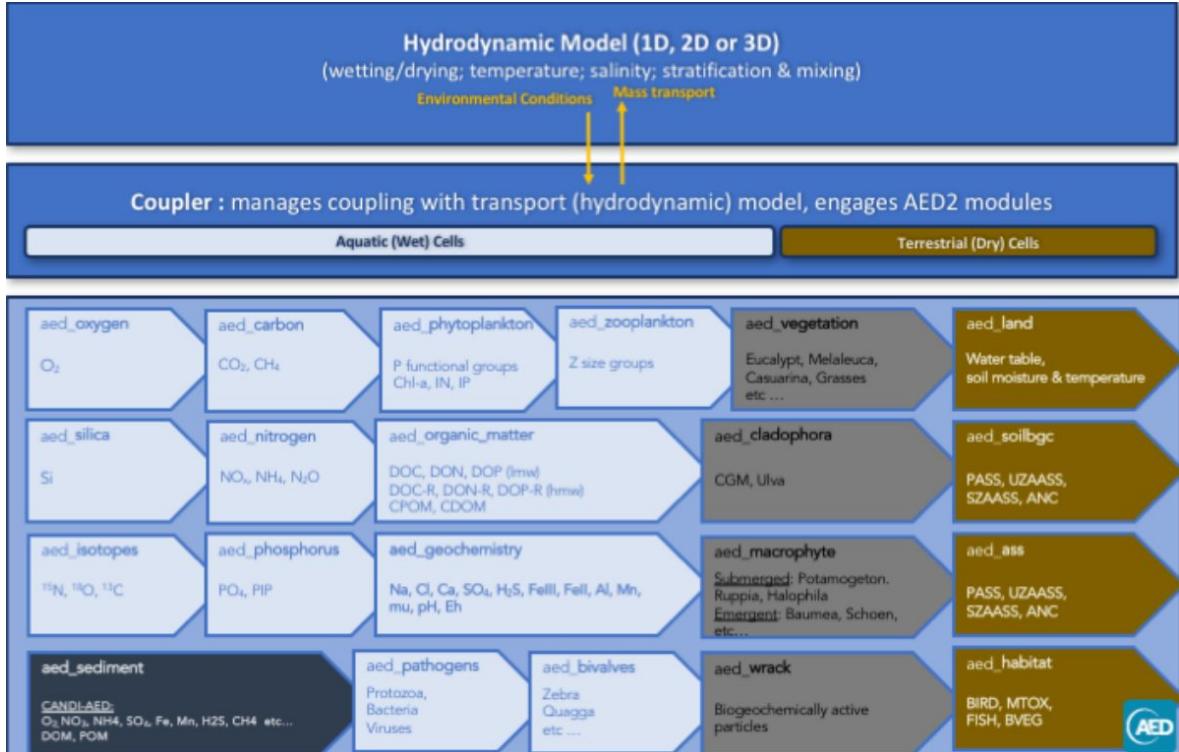
# AED2: water quality model

- Conceptualize your individual ecosystem:
  - Oxygen, nutrients, phytoplankton
  - Sediment biogeochemistry
  - Benthic communities



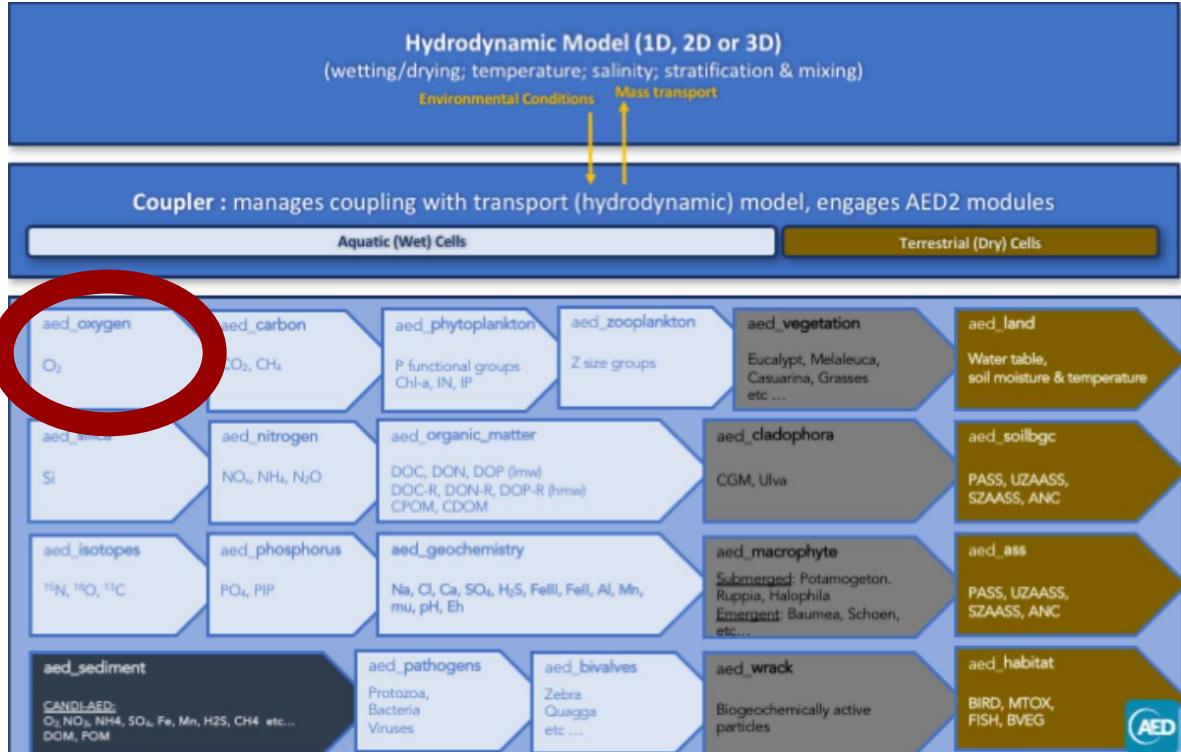
# AED2: modularised structure

- Interacts with GLM (two-way coupling)
- Array of modules are available
- Hierarchy of dependencies between modules, e.g., nitrogen reactions can depend on availability of oxygen



# AED2: modularised structure

- Interacts with GLM (two-way coupling)
- Array of modules are available
- Hierarchy of dependencies between modules, e.g., nitrogen reactions can depend on availability of oxygen

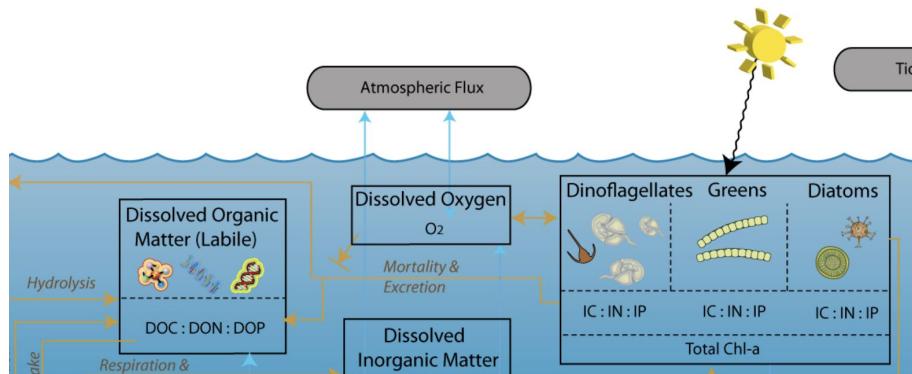
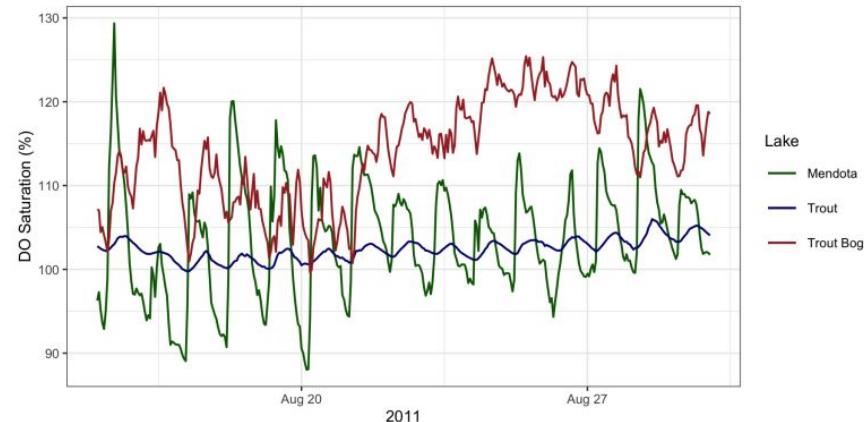


# AED2: dissolved oxygen

atmospheric exchange    sediment oxygen demand

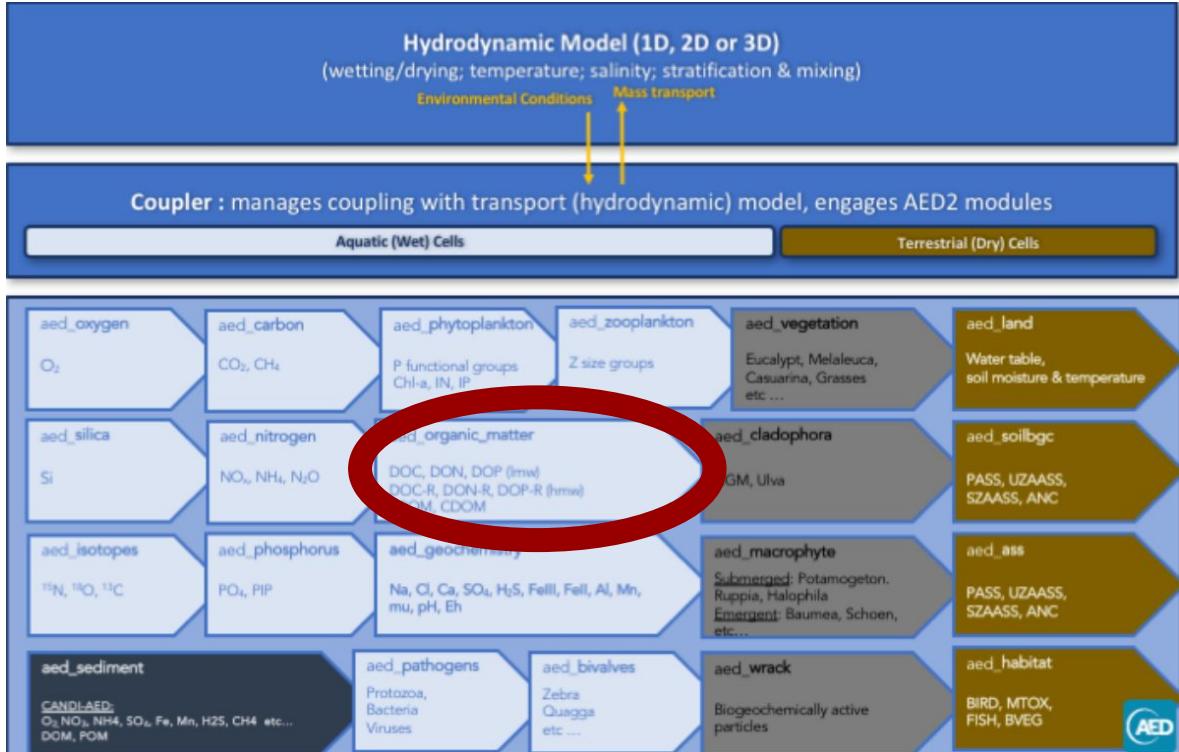
$$F_{[DO]}^{total} = k_{O_2} ([DO]_{air} - [DO]_{water}) + S_{SOD} f_{SOD}^T(T) f_{SOD}^{DO}(DO) + F_{mineralisation} + F_{nitrification} + F_{photosynthesis} + F_{respiration}$$

- Fast moving pool (minutes to days)
- Solved over vertical axis
- Plus advection and diffusion
- Photosynthesis/respiration = phytoplankton, zooplankton, seagrass, bivalves



# AED2: modularised structure

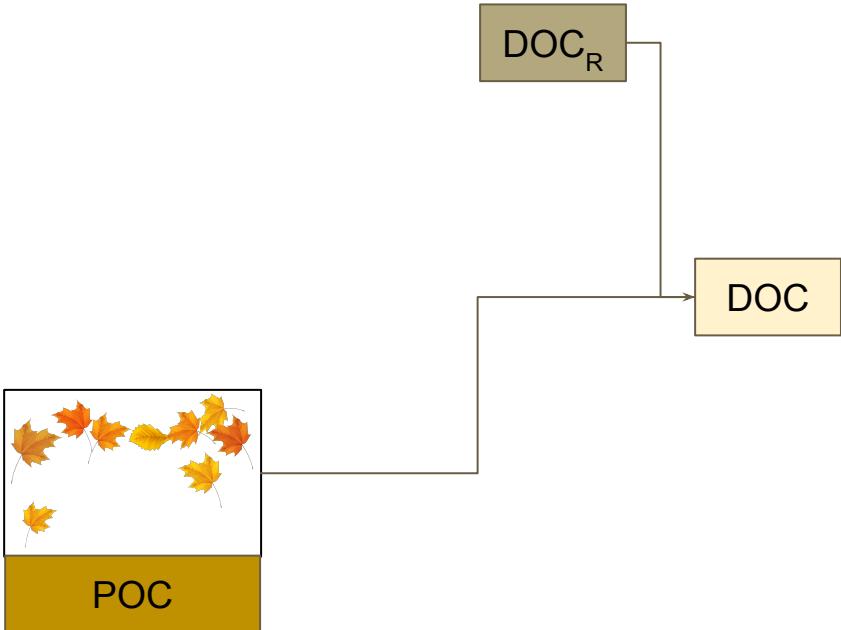
- Interacts with GLM (two-way coupling)
- Array of modules are available
- Hierarchy of dependencies between modules, e.g., nitrogen reactions can depend on availability of oxygen



# AED2: dissolved organic carbon - concept

Three pools:

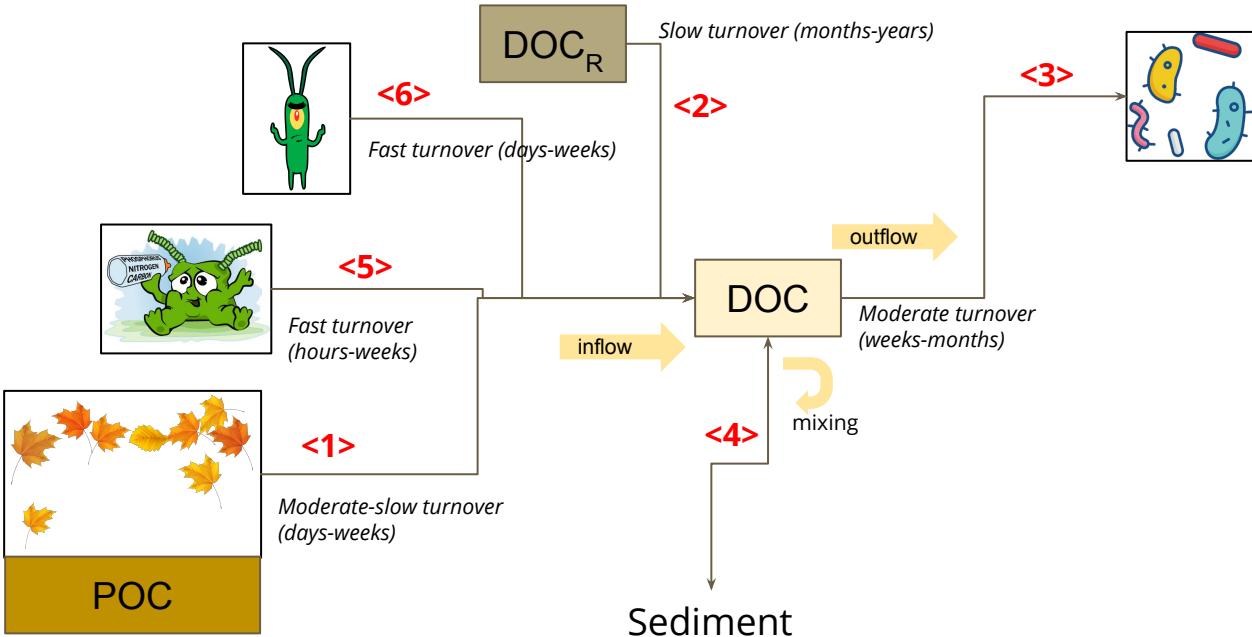
- Labile dissolved OC (**DOC**): fast turnover
- Refractory dissolved OC (**DOCr**): resistant to decomposition
- particulate OC (**POC**)



# AED2: dissolved organic carbon - concept

$$\frac{dDOC}{dt} = (1 - f_{ref})f_{decom}^{POC} + (1 - f_{photo})f_{photo}^{DOCR} - f_{miner}^{DOC} \pm f_{sed}^{DOC} + \sum_a N_{PHY} f_{excr}^{PHY-C_a} + \sum_z N_{ZOO} f_{excr}^z$$

<1>      <2>      <3>    <4>    <5>    <6>

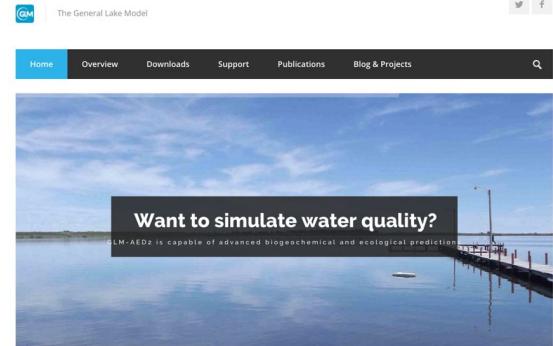


- Boxes are variables (states) you can observe
- In the model, C exists only in the boxes
- Arrows are fluxes, transformations, processes you generally cannot observe directly
- Fluxes determine how the boxes change through time
- Fluxes have equations with parameters
- Recalcitrant means the arrow is slow, labile fast
- Fitting a model requires changing parameter values so that boxes match observations, given boundary conditions
- All of these processes happen in all layers

# Running GLM (in R)

# Setting up GLM

- Installing GLM:
  - Download software package via AED GLM website:  
[http://aed.see.uwa.edu.au/research/models/GLM/latest\\_release.html](http://aed.see.uwa.edu.au/research/models/GLM/latest_release.html) (precompiled binaries)
  - Source code and compilation instructions are available, accessible via the GitHub repository -  
<https://github.com/AquaticEcoDynamics/GLM>
  - GLM3r in R  
devtools::install\_github("GLEON/GLM3r")



The screenshot shows the 'Downloads' section of the GLM website. It includes three circular icons: 'Science Basis' (describing GLM as a 1-dimensional lake water balance and stratification model), 'Suitability' (noting its use for natural and engineered lakes, shallow and deep), and 'Open Access' (mentioning it's an open-source project). Below these are sections for 'Latest Release' (GLM V3.0 (2019 release)), 'Download Model', and 'Source Code'. A note at the bottom states: '✓ GLM: The above GLM binary comes packaged with FABM & AED2. ✓ GLM+: Note the GLM+ version is GLM coupled with AED2 & AED2+. This is made available on request.'

# Setting up GLM

- Main file: glm3.nml
- Ordered into blocks
  - Begin with: &
  - End with: /
- Includes model setup and relative paths to water quality and driver data
- GLM executable needs the setup file

```
1 !-----  
2 ! general model setup  
3 !-----  
4 &glm_setup  
5   sim_name = 'Mendota'  
6   max_layers = 75  
7   min_layer_vol = 0.1  
8   min_layer_thick = 0.15  
9   max_layer_thick = 1.5  
10  density_model = 1  
11 /  
12 &mixing  
13   surface_mixing = 1  
14   coef_mix_conv = 0.2  
15   coef_wind_stir = 0.23  
16   coef_mix_shear = 0.3  
17   coef_mix_turb = 0.51  
18   coef_mix_KH = 0.3  
19   coef_mix_hyp = 0.4689587  
20   deep_mixing = 2  
21   diff = 0  
22 /  
23 &light  
24   light_mode = 0  
25   n_bands = 4  
26   light_extc = 1, 0.5, 2, 4  
27   energy_frac = 0.51, 0.45, 0.035, 0.005  
28   Benthic_Imin = 10  
29   Kw = 0.4315141  
30 /  
31 !-----  
32 ! water quality setup  
33 ! if this block is read, water quality functionality will be enabled  
34 !-----
```

# Setting up GLM

- Define the &morphometry
  - Area-depth relationship (&H and &A)
- Define what you want to simulate
  - State variables (temperature, salinity, etc.)
  - Grid resolution and time step (&glm\_setup, &time)
- Define external environment
  - Boundary conditions (inflows, meteorology, outflows) (&meteorology, &inflows, &outflows)
- Provide an initial condition (&init\_profiles)
- Start your simulation

## glm3.nml file

&glm_setup:	General simulation info
&wq_setup:	Water quality
&time:	Time control
&morphometry:	Lake morphometric information
&output:	Output file details
&init_profiles:	Initial vertical profiles
&meteorology:	Surface forcing and meeorology
&inflows:	Information about inflowing rivers
&outflows:	Information about outflows
&light:	Information about light climate

# Setting up GLM

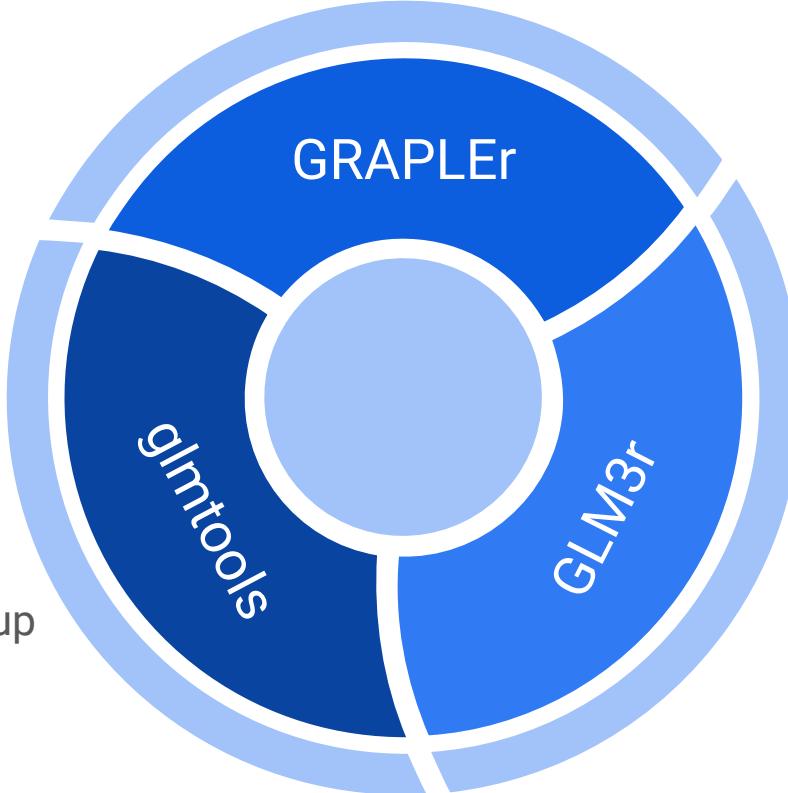
- Boundary conditions: data that drive the model
- At minimum, GLM needs meteorological data (hourly or daily) in CSV-format

```
Date,ShortWave,LongWave,AirTemp,RelHum,WindSpeed,Rain,Snow  
2009-01-01 00:00:00,0,191.50999,-15.609991,88.86569,2.9134343,0,0  
2009-01-01 01:00:00,0,217.56,-14.750006,88.178638,3.5679266,0,0  
2009-01-01 02:00:00,0,217.56,-14.15,88.305945,3.9796608,0,0  
2009-01-01 03:00:00,0,217.57001,-13.560004,88.286784,4.3862171,0,0  
2009-01-01 04:00:00,0,230.42999,-12.959998,88.028059,4.8016873,0,0  
2009-01-01 05:00:00,0,230.42999,-12.449988,87.745015,5.3791169,0,0  
2009-01-01 06:00:00,0,230.42999,-11.929999,87.288882,5.9664732,0,0  
2009-01-01 07:00:00,0,246.75,-11.41001,86.734669,6.5488395,0,0  
2009-01-01 08:00:00,0,246.75999,-10.620001,86.510658,6.995177,0,0  
2009-01-01 09:00:00,50.312,246.75999,-9.8200134,85.935506,7.4708298,0,0  
2009-01-01 10:00:00,168.256,253.35001,-9.0199951,85.14882,7.9805386,0,0  
2009-01-01 11:00:00,255.60001,253.35001,-7.8599915,85.236462,8.1806115,0,0  
2009-01-01 12:00:00,325.892,253.35001,-6.6900085,84.658597,8.4221193,0,0  
2009-01-01 13:00:00,349.651,265.15002,-5.5300049,83.680197,8.6924564,0,0  
2009-01-01 14:00:00,318.17099,265.14999,-4.9799866,84.281007,8.1392691,0,0  
2009-01-01 15:00:00,264.33701,265.14999,-4.4299988,84.709775,7.6592037,0,0  
2009-01-01 16:00:00,153.472,271.73001,-3.880011,84.970459,7.2817926,0,0  
2009-01-01 17:00:00,51.591999,271.73001,-4.1399902,86.532806,6.6806363,0,0
```

Time, FLOW, TEMP, SALT
2009-01-01 00:00:00,3.560861,-0.526716,0
2009-01-02 00:00:00,6.162997,-0.585426,0
2009-01-03 00:00:00,3.826635,-0.639793,0
2009-01-04 00:00:00,5.397271,-0.696168,0
2009-01-05 00:00:00,5.417987,-0.742644,0
2009-01-06 00:00:00,3.886593,-0.784772,0
2009-01-07 00:00:00,5.806259,-0.821455,0
2009-01-08 00:00:00,5.65432,-0.85336,0
2009-01-09 00:00:00,4.725836,-0.873803,0
2009-01-10 00:00:00,5.821722,-0.906718,0
2009-01-11 00:00:00,5.724123,-0.926215,0
2009-01-12 00:00:00,5.495776,-0.936452,0
2009-01-13 00:00:00,5.473368,-0.947798,0
2009-01-14 00:00:00,5.513825,-0.955154,0

- And flow data (inflow and/or outflow, hourly or daily) in CSV-format

# GLM in R



- Changing model setup
- Visualizing results
- Physical derivatives
- Calibration

- Distributed computing system
- Run multiple GLM simulations

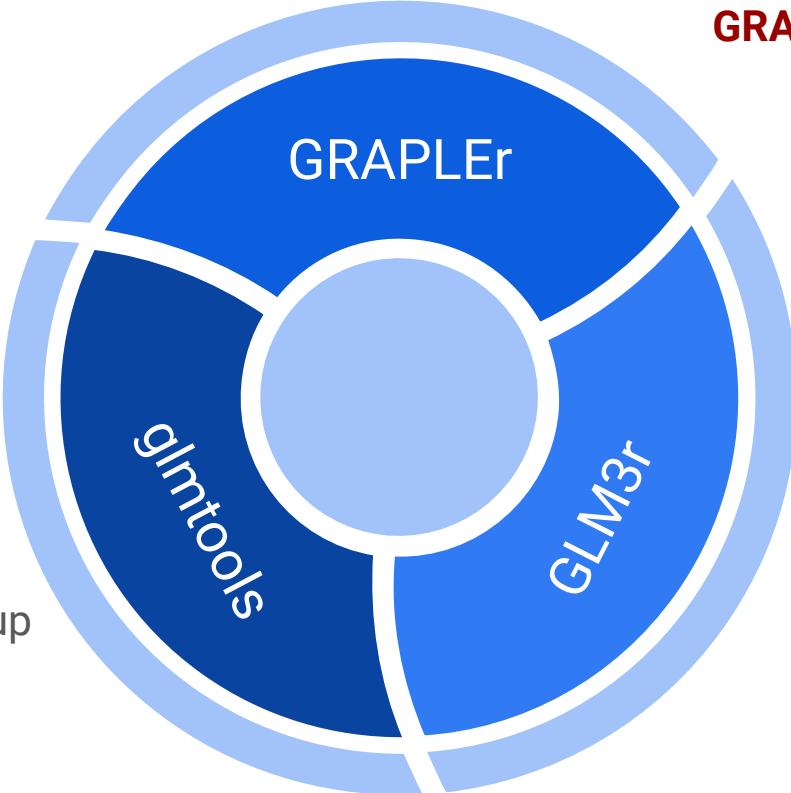
- Run GLM in R

# GLM in R



## USGS-R/glmtools

- Changing model setup
- Visualizing results
- Physical derivatives
- Calibration



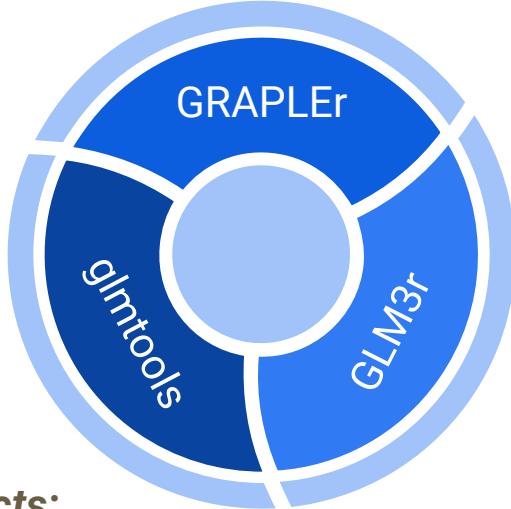
- Distributed computing system
- Run multiple GLM simulations

## GRAPLE/GRAPLEr

## GLEON/GLM3r

- Run GLM in R

# GLM in R



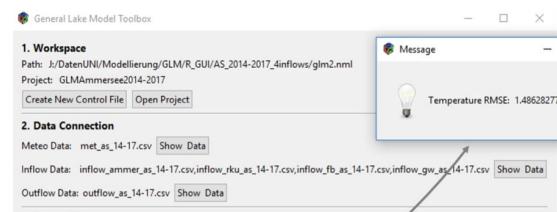
*More awesome R-related GLM projects:*

[Macrosystems EDDIE](#): classroom modules to teach macrosystems ecology

[LakeEnsemblR](#): R-package to run lake ensembles (release soon)

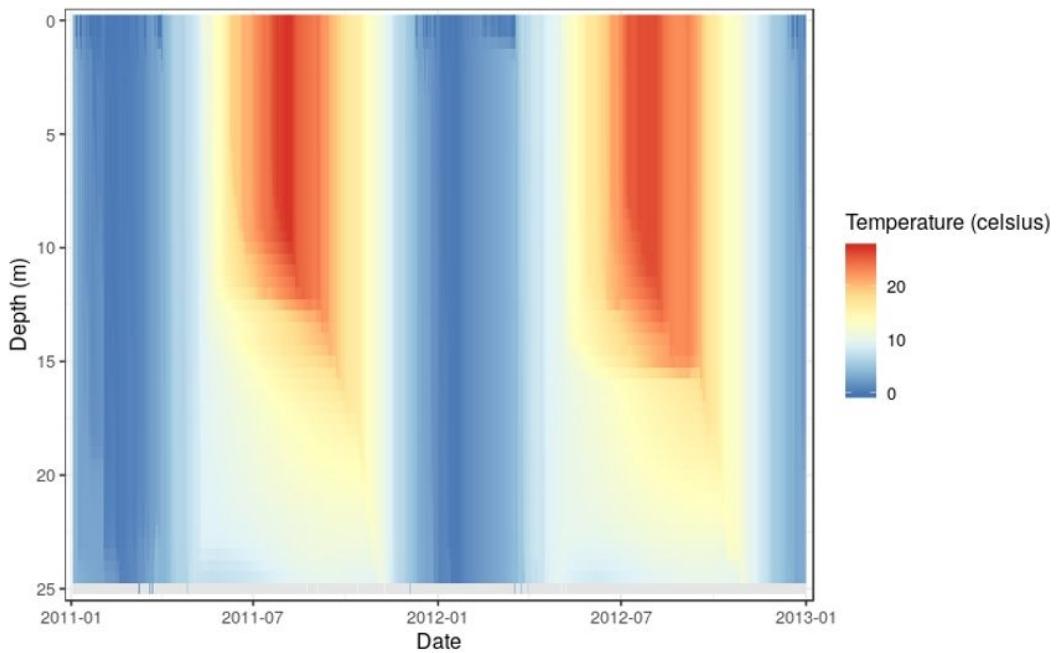
[glmGUI](#): GUI and toolbox to run GLM

[Columbus4Limnology](#): Run GLM online in the cloud (release soon)



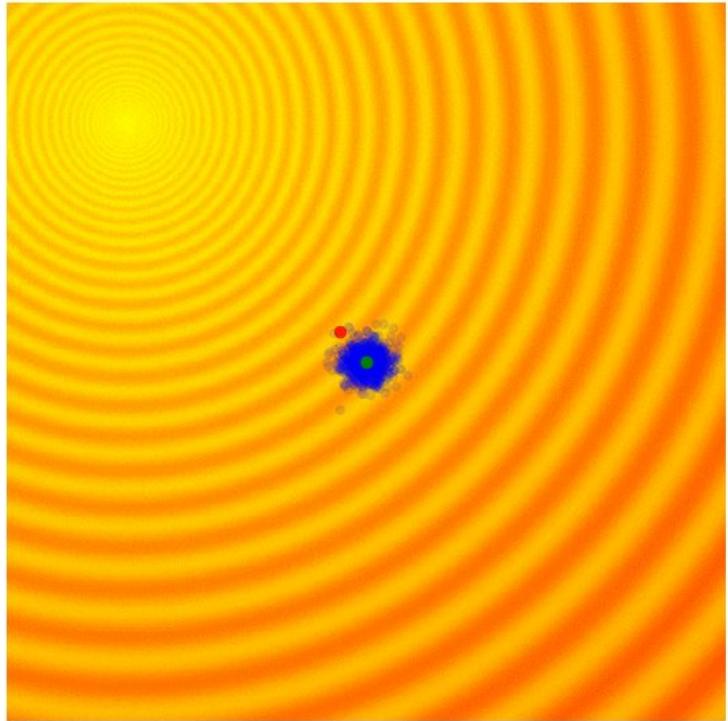
# glmtools

- Either from github or GRAN
- Offers lots of functionalities:
  - Manipulate model setup
  - Retrieve output data
  - Compare observed to simulated data
  - Visualize results
  - Calibration functions
- 'ggplot\_overhaul' branch includes up-to-date version with experimental features (e.g., water quality calibration)



# glmtools: calibration

- a) CMA-ES: derivative-free optimization, which adaptively changes the search path for every generation (modifying normal distribution and covariance matrix), covers wide search space, converges fast but needs more computational time ( $\sim O(N^2)$ )
- b) Nelder-Mead method: derivative-free optimization, direct search method



Finally some



# Wrapping up

Questions, issues, problems & feedback?

Join the official GLM slack



Thanks for joining!

Have fun lake modeling!



rladwig2@wisc.edu



@hydrobert