1D-AEMpy v0.1a - Technical Documentation

Robert Ladwig

2024-01-09

Technical documentation of the numerical implementation of the one-dimensinal Aquatic Ecosystem Model in Pyton (1D-AEMpy). This documentation describes the different modules that govern hydrodynamic and water quality calculations to simulate the vertical dynamics of water temperature, dissolved oxygen, phytoplankton biomass, nutrient concentration, and aspects of organic carbon.

Overview

This documentation describes the current version 0.1a of 1D-AEMpy, which includes hydrodynamic calculations coupled to a functional aquatic ecosystem model that explicitly simulates dissolved oxygen, phytoplankton biomass, a nutrient concentration (simulating reactive phosphorus), labile particulate organic carbon (POC-l), labile dissolved organic carbon (DOC-l), refractory particulate organic carbon (POC-r), and refractory dissolved organic carbon (DOC-r). The coupling between both models depends primarily on the light extinction coefficient (function of the sum of organic carbon concentrations) to the heat attenuation of the water column, as well as the derived eddy diffusivity governing the vertical diffusion of dissolved water quality variables.

> **Note**
>
> Version 0.1a differs to v0.1 in a set of important aspects:
>
> - v0.1 only simulates POC-l, DOC-l, POC-r, and DOC-r for water quality (meaning it essentially only captures the main metabolism reactions)
> - this means that in v0.1, GPP is a not dependent on any other state variable, but only on environmental conditions (light, nutrients)
> - v0.1 uses a time-dependent boundary condition for the internal availability of nutrients, i.e., TP
> - v0.1 does not include external mass fluxes, e.g., for POC-r and POC-l

1D-AEMpy iteratively runs 8 modules to update the states of a set of model variables.

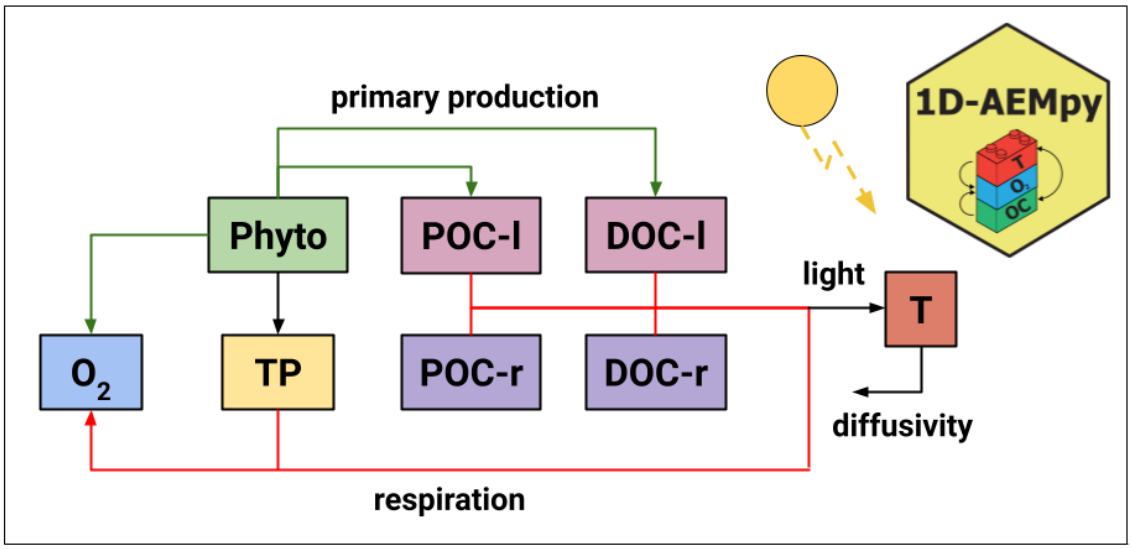

Figure 1: Conceptual diagram of 1D-AEMpy.

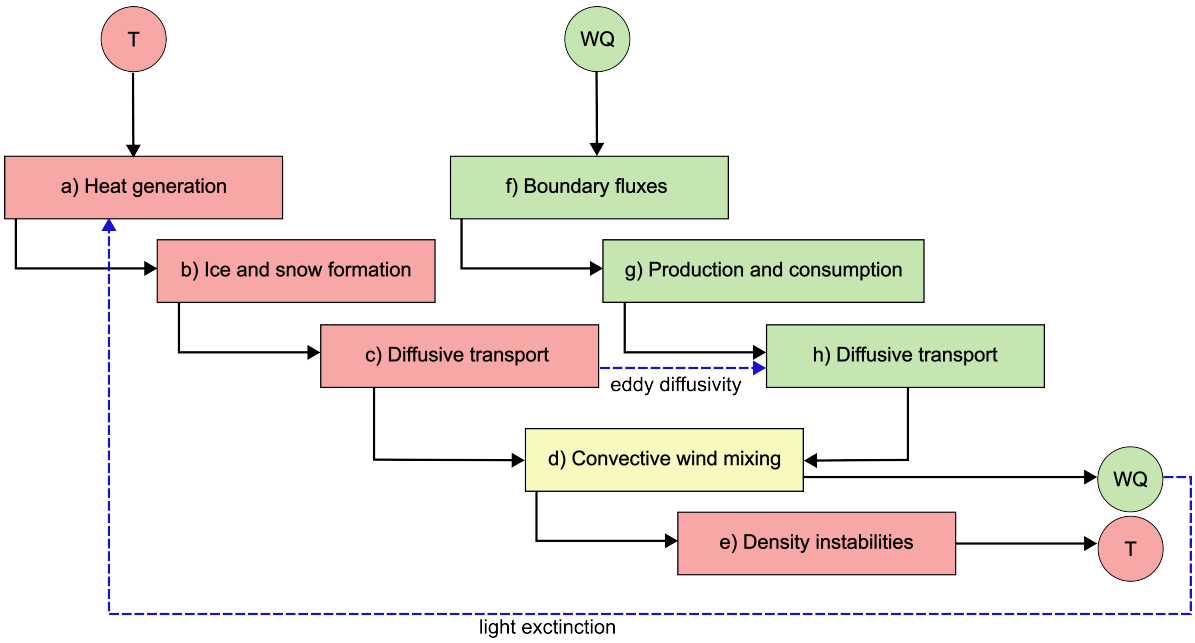

Figure 2: Modularization of 1D-AEMpy.

Hydrodynamics

A one-dimensional hydrodynamic lake model was developed to simulate the temperature, heat flux and stratification dynamics in a lake. The algorithms are based on the eddy diffusion approach *sensu* (Henderson-Sellers (1985)) and the MyLake (Saloranta and Andersen (2007)) model. Using the one-dimensional temperature diffusion equation for heat transport, we neglected any inflows and outflows, mass losses due to evaporation and water level changes:

$$\frac{\partial h}{\partial t} = 0$$

$$A\frac{\partial T}{\partial t} = \frac{\partial}{\partial z}(AK_z\frac{\partial T}{\partial z}) + \frac{1}{\rho_w c_p}\frac{\partial H(z)}{\partial z} + \frac{\partial A}{\partial z}\frac{H_{geo}}{\rho_w c_p}$$

where h is the water level (m), A is lake area (m2), T is water temperature (°C), t is time (s), K_z is the vertical diffusion coefficient m2 s-1, H is internal heat generation due to incoming solar radiation (W m2), ρ_w is water density (kg m-3), c_p is specific heat content of water (J kg-1 °C-1), and H_{geo} is internal geothermal heat generation (W m-2) (which was set to 0.1 W m-2 (Goudsmit et al. (2002))). Internal heat generation is implemented based on Beer-Lambert law for attenuation of short-wave radiation as a function of a constant light attenuation coefficient:

$$H(z) = (1-\alpha)\, I_s \exp\left(-k_d z\right)$$

where α is the albedo (−), I_s is total incident short-wave radiation (W m-2), and k_d is a light attenuation coefficient (m-1). For the boundary conditions, we assume a Neumann type for the temperature diffusion equation at the atmosphere-surface boundary, and a zero-flux Neumann type at the bottom:

$$\rho_w c_p (K_z \frac{\partial T}{\partial z})_{surface} = H_{net}$$

$$K_z (\frac{\partial T}{\partial z})_{bottom} = 0$$

where H_{net} is the net heat flux exchange between atmosphere and water column (W m-2). The neat heat flux exchange consisted of four terms:

$$H_{net} = H_{lw} + H_{lwr} + H_v + H_c$$

where H_{lw} is the incoming long-wave radiation (W m-2), H_{lwr} is emitted radiation from the water column (W m-2), H_v is the latent heat flux (W m-2), and H_c is the sensible heat flux (W m-2). Incoming and outgoing long-wave heat fluxes were derived using the formulations from Livingstone and Imboden (1989) and Goudsmit et al. (2002). The latent and sensible heat fluxes were calculated taking into account atmospheric stability using the algorithm by Verburg and Antenucci (2010).

The calculation of a temperature profile at every time step is modularized into four steps: (a) heat generation from boundary conditions, (b) ice and snow formation, (c) vertical diffusion, (d) wind-induced mixing, and (e) convective overturn. The one-dimensional temperature diffusion equation was discretized using the implicit Crank-Nicolson scheme (Press et al. (2007)), which being second-order in both space and time allows the modeling time step to be dynamic without numerical stability issues. The model was implemented in Python 3.7 with a default time step of $\Delta t = 3,600$ s and a default spatial discretization of $\Delta z = 0.5$ m.

a) Heat generation from boundary conditions

In the first step, the heat fluxes H, H_{geo} and H_{net} are applied over the vertical water column.

b) Ice, snow, and snow ice formation

In the second step, the ice and snow cover algorithm from MyLake @(saloranta_mylakemulti-year_2007) was applied to the model. Whenever water temperatures were equal or below the freezing point of water (set to 0 °C), ice formation was triggered. All layers with water temperatures below the freezing point were set to 0 °C, and the heat deficit from atmospheric heat exchange was converted into latent heat of ice formation. Stefan's law was applied to calculate ice thickness when air temperatures were below freezing point triggering ice formation (e.g., Leppäranta (1993)):

$$h_{ice} = \sqrt{h_{ice}^2 + \frac{2\kappa_{ice}}{\rho_{ice} L}(T_f - T_{ice})\Delta t}$$

where h_{ice} is ice thickness (m), κ_{ice} is thermal conductivity of ice $(W\ K^{-1}\ m^{-1})$, ρ_{ice} is ice density $(kg\ m^{-3})$, L is latent heat of freezing $(J\ kg^{-1})$, T_f is water temperature at freezing point $(T_f = 0\ ^\circ C)$, and T_{ice} is the temperature of the ice surface. The formation of a snow layer on top of the ice layer depended on the amount of precipitation. Further, whenever the weight of snow exceeded the buoyancy capacity of the ice layer, enough water to offset the exceedance forms a snow ice layer with the same properties as ice. When air temperatures were above the freezing point, ice and snow growth ceased, and snow and ice melting were initiated with ice melt requiring no snow to exist. Here, total energy of melting was taken from the total heat flux H_{net}. Once the ice layer has disappeared, the default model routine continued. For more details, we refer the reader to Saloranta and Andersen (2007).

c) Vertical (turbulent) diffusion

In the third step, vertical turbulent diffusion between adjacent grid cells was calculated. Here, we applied a centered difference approximation for temperature at the next time step. The

vertical turbulent diffusion coefficients, K_t, were calculated using an empirical relationship depending on the Richardson number:

$$K_z = \frac{K_0}{(1 + 5Ri(z))^2} + K_m$$

where K_0 is an adjustable parameter (set to 10^{-2} m s-2) R_i is the Richardson number, and K_m is the background eddy diffusivity (Pacanowski and Philander (1981), Jabbari et al. (2023)). The Richardson number was quantified as:

$$R_i = \frac{-1 + [1 + 40N^2k^2z^2 / \left(w^{*2}\exp(-2k^*z)\right)]^{(1/2)}}{20}$$

with k as the Karman constant ($k = 0.4$), and the squared buoyancy frequency, $N^2 = \frac{g}{\rho_w}\frac{\partial \rho_w}{\partial z}$ (s^{-2}) (Henderson-Sellers (1985)). Friction velocity w^* was calculated as:

$$w^* = C_D U_2$$

where the drag coefficient C_D was set to 1.3 x 10^{-3}, and U_2 is the wind speed at 2 m above surface (m s-1). All values of N^2 less than 7.0 x 10^{-5} s-2 were set to 7.0 x 10^{-5} s-2 (Hondzo and Stefan (1993)).

To replicate a lag in the mixing dynamics, we set the values of K_z to the average between the current profile and the one from the previous time step (Piccolroaz and Toffolon (2013)).

Numerical implementation

The implicit Crnak-Nicolson scheme, whichis second-order derivative in space and time, was applied to solve the one-dimensional temperature transport equation for the diffusive transport. Here, we average the response in space between the current and the next time step:

$$A\frac{\partial T}{\partial t} = AK_z\frac{\partial T}{\partial z}$$

$$A\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{1}{2}[AK_z\frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{\Delta z^2} + AK_z\frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta z^2}]$$

Here, we can apply $\alpha = AK_z\frac{\Delta t}{\Delta z^2}$, transforming the equation to:

$$-\frac{\alpha}{2}T_{i+1}^{n+1} + (A + \alpha)T_i^{n+1} - \frac{\alpha}{2}T_{i-1}^{n+1} = \frac{\alpha}{2}T_{i+1}^n + (A - \alpha)T_i^n + \frac{\alpha}{2}T_{i-1}^n$$

$$-\frac{\alpha}{2}T_{i+1}^{n+1} + (A + \alpha)T_i^{n+1} - \frac{\alpha}{2}T_{i-1}^{n+1} = R_i^n$$

where the right-hand side quantity R_i^n is known at the beginning of each time step.

In the current version, we asusme static Dirichlet boundary conditions for the diffusive transport, which transforms the equation into a matrix (given here for an example for 5 times 5 rows and columns):

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -\frac{\alpha}{2} & 1+\alpha & -\frac{\alpha}{2} & 0 & 0 \\ 0 & -\frac{\alpha}{2} & 1+\alpha & -\frac{\alpha}{2} & 0 \\ 0 & 0 & -\frac{\alpha}{2} & 1+\alpha & -\frac{\alpha}{2} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_1^{n+1} \\ T_2^{n+1} \\ T_3^{n+1} \\ T_4^{n+1} \\ T_5^{n+1} \end{bmatrix} = \begin{bmatrix} T_1^n \\ R_2^n \\ R_3^n \\ R_4^n \\ T_5^n \end{bmatrix}$$

d) Wind-induced mixing

To ensure an adequate representation of the effects of wind-induced mixing, which was not incorporated into the turbulent diffusive transport step, an additional step based on the concept of integral energy was applied following the algorithms by Saloranta and Andersen (2007) and Ford and Stefan (1980). Generally, the avaible external turbulent kinetic energy (TKE) is compared to the potential energy of the water column that is needed to lift up denser water from below a mixed layer into a newly formed mixed layer until the external TKE is depleted. TKE was quantified as:

$$TKE = C_{shelter} A_s \sqrt{\frac{\tau^3}{\rho_w}} \Delta t$$

where $C_{shelter}$ is a wind-sheltering coefficient, and τ is the wind shear stress. $C_{shelter}$ was parameterized based on Hondzo and Stefan (1993):

$$C_{shelter} = 1.0 - exp(-0.3 A_s)$$

TKE is compared with the available potential energy (PE) in the water column:

$$PE = g \Delta \rho_w \frac{V_{mixed} V_z}{V_{mixed} + V_z} (z_{mixed} + \Delta z_{M,z} - z_{M,mixed})$$

where $\Delta \rho_w$ is the density difference between layer z and the mixed-layer (epilimnion) density, V_{mixed} is the mixed-layer volume, and $\Delta z_{M,z}$ is the distance from layer z's center of mass, z_M, to the bottom of the mixed-layer.

If TKE $\geq$ PE, then layer z will be mixed into the mixed layer (the epilimnion). This also includes water quality variables, which will be volume-averaged. Once TKE $<$ PE, the remaining energy will be used for partial mixing as in Saloranta and Andersen (2007).

e) Convective overturn

In the final step, any density instabilities over the vertical water column were mixed with the first stable layer below an unstable layer. Here, we applied the area weighed mean of

temperature between two layers to calculate the new temperature of the previously unstable grid cell. Density differences between two layers were averaged until the difference was equal or less than 1 x 10^{-3} kg m-3. {An arbitrary cutoff of 1 x 10^{-3} kg m-3 was chosen to reduce computational run time by preventing averaging of density profiles to the fourth decimal point.}

Water quality

The current implementation simulates dissolved oxygen, phytoplankton biomass, a nutrient concentration (simulating reactive phosphorus), labile particulate organic carbon (POC-l), labile dissolved organic carbon (DOC-l), refractory particulate organic carbon (POC-r), and refractory dissolved organic carbon (DOC-r) following a general equation of:

$$A\frac{\partial C}{\partial t} + w\frac{\partial C}{\partial z} - \frac{\partial}{\partial z}(AK_z\frac{\partial C}{\partial z}) = P(C) - D(C)$$

where w is a sinking rate, $P(C)$ is a production term, and $D(C)$ is a consumption (destruction) term.

The main feedback of water quality to hydrodynamics is via the light extinction coefficient:

$$k_d = k_{d,water} + k_{d,DOC}\overline{C_{DOC}} + k_{d,POC}\overline{C_{POC}}$$

where there are specific light extinction coefficients for background water, DOC and POC, $k_{d,water}$, $k_{d,DOC}$ and $k_{d,POC}$, respectively, as well as the average sum of DOC and POC concentrations, $\overline{C_{DOC}}$ and $\overline{C_{POC}}$, respectively, from the surface to the mean depth of the lake.

The water quality simulation is split into three modules dealing with (f) effects of external and internal boundaries, (g) internal production and conumption pathways, and (h) vertical transport.

f) Boundary fluxes

Boundary fluxes include mass addition from external loadings, gas-exchange between the atmosphere and the water column, as well as sediment consumption fluxes.

For mass fluxes, external loadings in m s-1 are added directly to the layer adjacent to the atmosphere-water interface and are multiplied with the time step.

Surface gas exchange for dissolved oxygen was parameterized as

$$F_{atm} = k_p(C_{DO,atm} - C_{DO})A\Delta t$$

where k_p is the piston velocity, and $C_{DO,atm}$ is the saturated oxygen concentration. The piston velocity was calculated using the empirical gas exchange model following Vachon and Prairie (2013) based on lake area and wind velocity.

Sediment oxygen demand for the consumption of the dissolved oxygen conentration C_{DO} (a sink for oxygen in the model) is quantified as

$$SOD = (F_{DO} + \frac{D_{diff}}{\delta} C_{DO} A)\theta_R^{T-20}\Delta t$$

where F_{DO} is an idealized area flux rate (g m-2 s-1), D_{diff} is the molecular oxygen diffusion coefficient (m2 s-1), δ is the thickness of the sediment diffusive boundary layer (m, set to a constant value of 0.001 m), and θ is a temperature multiplier, here for respiration R. D_{diff} was quantified as a function of water temperature following Han and Bartels (1996).

The release of nutrients from the sediment, SNR (a source of nutrients in the model), was parameterized as a function depending on dissolved oxygen availability (using Michaelis-Menten kinetics) and temperature:

$$SNR = (F_{NUTR} \frac{C_{DO}}{k_{DO} + C_{DO}} A)\theta_R^{T-20}\Delta t$$

where F_{NUTR} is an idealized area flux rate (g m-2 s-1), and k_{DO} is the half-saturation concentration of dissolved oxygen (g m-3).

> **Note**
>
> In v0.1, SOD is quantified using Michaelis-Menten kinetics:
>
> $$SOD = (F_{DO} \frac{C_{DO}}{k_{DO} + C_{DO}} A)\theta_R^{T-20}\Delta t$$
>
> Furter, GPP is quantified as a direct addition to oxygen, POC-l, and DOC-l at every time step using a forward Euler scheme:
>
> $$GPP = R_{molar}(Hp_1 I_P C_{nutr} \theta_{GPP}^{T-20})$$
>
> where R_{molar} is the respective molar ratio to convert from carbon units into something else, p_1 is the conversion from short-wave radiation to PAR (2.114), and I_P is a calibration coeffcient.
>
> $$M_{DO}^t = M_{DO}^{t-1} + \Delta t * GPP$$
> $$M_{POC-l}^t = M_{POC-l}^{t-1} + 0.2\Delta t * GPP$$
> $$M_{DOC-l}^t = M_{DOC-l}^{t-1} + 0.8\Delta t * GPP$$

g) Production and consumption

The model tracks the mass dynamics of dissolved oxygen, M_{DO}, phytoplankton biomass, M_{phyto}, nutrients, M_{nutr}, labile particulate organic carbon, M_{POC-l}, labile dissolved organic carbon, M_{DOC-l}, refractory particulate organic carbon, M_{POR-r}, and refractory dissolved organic carbon, M_{POC-l}. The respective concentrations, C were quantified by dividing the masses with the voume.

For production terms, gross primary production was quantified as

$$GPP = M_{phyto} R_{molar} (H p_1 I_P C_{nutr} \theta_{GPP}^{T-20})$$

where R_{molar} is the respective molar ratio to convert from carbon units into something else, p_1 is the conversion from short-wave radiation to PAR (2.114), and I_P is a calibration coeffcient.

Respiration was quantified as depending on oxygen availability

$$RSP = M_i R_{molar} R_i \frac{C_{DO}}{k_{DO} + C_{DO}} \theta_R^{T-20}$$

with M_i either being nutrient, POC-l, DOC-l, POC-r, or DOC-r, and R_i is the respective respiration rate.

Phytoplankton growth was quantified similar to GPP

$$G = p_2 H p_1 \frac{C_{phyto}}{k_{phyto} + C_{phyto}} \theta_{GPP}^{T-20}$$

with p_2 being the growth rate of algae (d-1).

The respective production and consumption terms for each variable are listed below:

$$P_{DO} - D_{DO} = GPP - (RSP_{POC-l} + RSP_{DOC-l} + RSP_{POC-r} + RSP_{DOC-r} + RSP_{nutr})$$

$$P_{POC-l} - D_{POC-l} = 0.8 GPP - RSP_{POC-l}$$

$$P_{DOC-l} - D_{DOC-l} = (RSP_{POC-l} + 0.2 GPP) - RSP_{DOC-r}$$

$$P_{POC-r} - D_{POC-r} = -RSP_{POC-r}$$

$$P_{DOC-r} - D_{DOC-r} = RSP_{POC-r} - RSP_{DOC-r}$$

$$P_{phyto} - D_{phyto} = G M_{phyto} - p_3 M_{phyto} \theta_{GPP}^{T-20}$$

with p_3 as a grazing rate (d-1).

$$P_{nutr} - D_{nutr} = p_3 p_4 M_{phyto} R_{molar} \theta_{GPP}^{T-20} - RSP_{nutr}$$

with p_4 as the grazing ratio (hence, how much of the grazed material is converted from the grazer into nutrients, similar to an excretion rate).

> **Note**
>
> In v0.1, no production terms are used as GPP is included in the boundary flux module. Therefore, the following consumption terms are used:

$$P_{DO} - D_{DO} = -(RSP_{POC-l} + RSP_{DOC-l} + RSP_{POC-r} + RSP_{DOC-r})$$

$$P_{POC-l} - D_{POC-l} = -RSP_{POC-l}$$
$$P_{DOC-l} - D_{DOC-l} = -RSP_{DOC-r}$$
$$P_{POC-r} - D_{POC-r} = -RSP_{POC-r}$$
$$P_{DOC-r} - D_{DOC-r} = -RSP_{DOC-r}$$

Numerical implementation

For internal production and consumption fluxes, we applied the 2nd-order Modified Patankar-Runge-Kutta scheme, which is mass conservative and unconditionally positive (Burchard, Deleersnijder, and Meister (2003)):

$$C_i^{(1)} = C_i^n + \Delta t(\sum_{j=1}^{I} P_{i,j}(C^n) - \sum_{j=1}^{I} D_{i,j}(C^n)\frac{C_i^{(1)}}{C_i^n})$$

$$C_i^{n+1} = C_i^n + \frac{\Delta t}{2}(\sum_{j=1}^{I}(P_{i,j}(C^n) + p_{i,j}(C^{(1)}))\frac{C_j^{n+1}}{C_j^{(1)}} - \sum_{j=1}^{I}(D_{i,j}(C^n) + D_{i,j}(C^{(1)}))\frac{C_j^{n+1}}{C_j^{(1)}})$$
$$\text{with } i = 1, ..., I$$

where i determines the current water quality constituent, P is a source term, and D is a sink term.

h) Transport

For vertical transport, sinking and diffusion were parameterized for particulate and dissolved substances, respectively. For phytoplankton biomass, sinking and diffusion were both applied.

Sinking was conceptualized as a sinking loss that was applied to every cell:

$$SL = M_i \frac{v_{i,settling}}{\Delta z}\Delta t$$

with $v_{i,settling}$ as a constant settling velocity. Similarily, a constant sedimentation rate, $v_{sedimentation}$ was applied to the grid cell adjacent to the sediment-water interface to remove mass from the model over time.

Turbulent diffusion was applied to the dissolved substances according to the previously described scheme in (c).

Model parameterization

Model parameter	Symbol	Description	Default value
u	T	Initial temperature conditions (C)	-
o2	M_{DO}	Initial oxygen conditions (g)	-
docr	M_{DOC-r}	Initial DOC-r conditions (g)	-
docl	M_{DOC-l}	Initial DOC-l conditions (g)	-
pocr	M_{POC-r}	Initial POC-r conditions (g)	-
pocl	M_{POC-l}	Initial POC-l conditions (g)	-
alg	M_{phyto}	Initial phytoplankton conditions (g)	-
nutr	M_{nutr}	Initial nutrient conditions (g)	-
startTime	-	Start time-date (s)	-
endTime	-	End time-date (s)	-
area	A	Depth-discrete areas (m2)	-
volume	V	Depth-discrete volumes (m3)	-
depth	z	Depth-discrete depths (m)	-
zmax	-	Maximum lake depth	-
nx	-	Maximum grid cell number	50
dt	Δt	Time step (s)	3600
dx	Δz	Space step (s)	0.5
daily_meteo	-	Meteorological driver data	-
secview	-	Observed Secchi disk data (ignored since v0.1)	-
phosphorus_data	-	Observed phosphorus data (ignored in v0.1a)	-
ice	-	Initial ice condition, TRUE or FALSE	FALSE
Hi	h_{ice}	Initial ice thickness (m)	0
Hs	-	Initial snow thickness (m)	0
Hsi	-	Initial snow ice thickness (m)	0
iceT	-	Initial moving average temperature for ice (C)	6
supercooled	-	Initial amount of layers below freezing point	0
diffusion_method	-	Numerical method for K_z	“pacanowskiPhilander”
scheme	-	Numerical method for diffusion	“implicit”
km	K_m	Background eddy diffusivity (m2s-1)	$1.4 * 10^{-7}$

Model parameter	Symbol	Description	Default value
k0	-	Adjustable eddy diffusivity parameter (m2s-1)	$1.0 * 10^{-2}$
weight_kz	-	Weighting of eddy diffusivity	0.5
kd_light	k_d	Light attenuation (m-1, ignored since v0.1)	0.6
densThresh	-	Density cutoff (kg m-3)	$1.0 * 10^{-2}$
albedo	α	Albedo	0.1
eps	-	Emissivity of water	0.97
emissivity	-	Emissivity of water (same as eps)	0.97
sigma	-	Stefan-Boltzmann constant (W m-2K-4)	$5.67 * 10^{-8}$
sw_factor	-	Multiplier for short-wave radiation	1.0
wind_factor	-	Multiplier for wind speed	1.0
at_factor	-	Multiplier for air temperature	1.0
turb_factor	-	Multiplier for turbulent heat fluxes	1.0
p2	-	Ignored since v0.1	1.0
B	-	Ignored since v0.1	0.61
g	g	Gravitational acceleration (m s-2)	9.81
Cd	C_D	Momentum coefficient of wind	0.0013
meltP	-	Multiplier for ice melting	1.0
dt_iceon_avg	-	Ratio of ice forming temperature	0.8
Hgeo	H_{geo}	Geothermal heat flux (W m-2)	0.1
KEice	-	TKE multiplier for ice conditions (ignored since v0.1)	0
Ice_min	-	Minimum ice thickness (m)	0.1
pgdl_mode	-	Additional data processing	"on"
rho_snow	ρ_{snow}	Initial snow density (kg m-3)	250
p_max	-	Multiplier for GPP (ignored since v0.1)	1/86400
IP	I_P	Calibration coefficient for GPP	$3.0 * 10^{-6}/86400$
theta_npp	θ_{GPP}	Arrhenius temperature multiplier	1.0
theta_r	θ_R	Arrhenius temperature multiplier	1.08
conversion_constant		Multiplier for GPP (ignored since v0.1)	$1.0 * 10^{-4}$
sed_sink	F_{nutr}	Idealized nutrient source rate (g m-2 s-1)	0.01/86400
k_half	k_{DO}	Half-saturation concentration (g m-3)	0.5
resp_docr	-	DOC-r respiration rate (s-1)	0.008/86400
resp_docl	-	DOC-l respiration rate (s-1)	0.008/86400

Model parameter	Symbol	Description	Default value
resp_pocr	-	POC-r respiration rate (s-1)	0.08/86400
resp_pocl	-	POC-l respiration rate (s-1)	0.08/86400
grazing_rate	p_3	Grazing rate of phytoplankton (s-1)	$3.0 * 10^{-3}$/86400
pocr_settling_rate	$v_{POC-r,settling}$	POC-r settling rate (m s-1)	0.2/86400
pocl_settling_rate	$v_{POC-l,settling}$	POC-l settling rate (m s-1)	0.2/86400
algae_settling_rate	$v_{phyto,settling}$	Phytoplankton settling rate (m s-1)	$1.0 * 10^{-5}$/86400
sediment_rate	$v_{sedimentation}$	Sedimentation rate (m s-1)	0.5/86400
piston_velocity	k_p	Piston velocity for gas exchange (m s-1, ignored since v0.1)	1.0/86400
light_water	$k_{d,water}$	Background light extinction (m-1)	0.125
light_doc	$k_{d,DOC}$	DOC light extinction (m-1)	0.02
light_poc	$k_{d,POC}$	POC light extinction (m-1)	0.7
mean_depth	-	Mean depth of the lake (m)	$\frac{\sum V}{A_s}$
W_str	$C_{shelter}$	Manual wind-sheltering coefficient	"None"
tp_inflow	-	Boundary condition of total phosphorus (m-1, ignored in v0.1a)	-
pocr_inflow	-	Mass influx of POC-r (m-1, ignored in v0.1)	-
pocl_inflow	-	Mass flux of POC-l (m-1, ignored in v0.1)	-
f_sod	F_{DO}	Idealized oxygen sink rate (g m-2 s-1, ignored in v0.1)	0.1/86400
d_thick	δ	Thickness of diffusive boundary layer (m)	0.001
growth_rate	p_2	Phytoplankton growth ratio (s-1)	$1.0 * 10^{-3}$/86400
grazing_ratio	p_4	Grazing ratio	0.1

Burchard, Hans, Eric Deleersnijder, and Andreas Meister. 2003. "A High-Order Conservative Patankar-Type Discretisation for Stiff Systems of Production-Destruction Equations." *Applied Numerical Mathematics.* 47 (1): 1–30. https://doi.org/https://doi.org/10.1016/S0168-9274(03)00101-6.

Ford, Dennis E, and Heinz G Stefan. 1980. "Thermal Predictions Using Integral Energy Model." *Journal of the Hydraulics Division* 106 (1): 39–55. https://doi.org/10.1061/JYCEAJ.0005358.

Goudsmit, G.-H., H. Burchard, F. Peeters, and A. Wüest. 2002. "Application of k-$\Epsilon$ Turbulence Models to Enclosed Basins: The Role of Internal Seiches." *Journal of Geophysical Research: Oceans* 107 (C12): 23-1-23-13. https://doi.org/10.1029/2001JC000954.

Han, Ping, and David M. Bartels. 1996. "Temperature Dependence of Oxygen Diffusion in H2O and D2O." *J. Phys. Chem.* 100 (13): 5597–5602. https://doi.org/https://doi.org/10.1021/jp952903y.

Henderson-Sellers, B. 1985. "New Formulation of Eddy Diffusion Thermocline Models." *Applied Mathematical Modelling* 9 (6): 441–46. https://doi.org/10.1016/0307-904X(85)90110-6.

Hondzo, Midhat, and Heinz G. Stefan. 1993. "Lake Water Temperature Simulation Model." *Journal of Hydraulic Engineering* 119 (11): 1251–73. https://doi.org/10.1061/(ASCE)0733-9429(1993)119:11(1251).

Jabbari, Aidin, Reza Valipur, Josef D. Ackerman, and Yerubandi R. Rao. 2023. "Nearshore-offshore exchanges by enhanced turbulent mixing along the north shore of Lake Ontario." *Journal of Great Lakes Research* 49 (3): 596–607. https://doi.org/https://doi.org/10.1016/j.jglr.2023.03.010.

Leppäranta, Matti. 1993. "A Review of Analytical Models of Sea-ice Growth." *Atmosphere-Ocean* 31 (1): 123–38. https://doi.org/10.1080/07055900.1993.9649465.

Livingstone, David M., and Dieter M. Imboden. 1989. "Annual Heat Balance and Equilibrium Temperature of Lake Aegeri, Switzerland." *Aquatic Sciences* 51 (4): 351–69. https://doi.org/10.1007/BF00877177.

Pacanowski, R. C., and S. G. H. Philander. 1981. "Parameterization of Vertical Mixing in Numerical Models of Tropical Oceans." *Journal of Physical Oceanography* 1: 1443–51.

Piccolroaz, Sebastiano, and Marco Toffolon. 2013. "Deep Water Renewal in Lake Baikal: A Model for Long-Term Analyses: Deep Water Renewal in Lake Baikal." *Journal of Geophysical Research: Oceans* 118 (12): 6717–33. https://doi.org/10.1002/2013JC009029.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. 2007. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press.

Saloranta, Tuomo M., and Tom Andersen. 2007. "MyLake—A Multi-Year Lake Simulation Model Code Suitable for Uncertainty and Sensitivity Analysis Simulations." *Ecological Modelling*, Uncertainty in Ecological Models, 207 (1): 45–60. https://doi.org/10.1016/j.ecolmodel.2007.03.018.

Vachon, Dominic, and Yves T. Prairie. 2013. "The Ecosystem Size and Shape Dependence of Gas Transfer Velocity Versus Wind Speed Relationships in Lakes." Edited by Ralph Smith. *Canadian Journal of Fisheries and Aquatic Sciences* 70 (12): 1757–64. https://doi.org/10.1139/cjfas-2013-0241.

Verburg, Piet, and Jason P. Antenucci. 2010. "Persistent Unstable Atmospheric Boundary Layer Enhances Sensible and Latent Heat Loss in a Tropical Great Lake: Lake Tanganyika." *Journal of Geophysical Research* 115 (D11): D11109. https://doi.org/10.1029/2009JD012839.