User Manual

TABLE of CONTENTS

1. Introduction

Advanced Lake Biogeochemistry Model (ALBM) is a one-dimensional process-based lake biogeochemistry model that was developed by Dr. Zeli Tan to predict the response of lake thermal and carbon dynamics and water quality to environmental changes (Tan et al., 2015, 2017, 2018). It consists of several modules including those for the radiative transfer, the water/sediment thermal circulation, the water/sediment biogeochemistry, phytoplankton biomass, and the gas diffusive and ebullition transportation. Although the model was originally developed for Arctic lakes (Tan et al., 2015), it has since been used for lakes in diverse environments (Guseva et al., 2020; Guo et al., 2020a, 2020b). ALBM is a member model of the Inter-Sectoral Impact Model Intercomparison Project (ISIMIP) lake sector (https://www.isimip.org/impactmodels/details/232/).

To make this user manual more accessible we will briefly describe the contents of each chapter. Chapter 2 "Scientific description" describes the physical and biogeochemical processes modeled in ALBM. Chapter 3 "Model setup" provides the list of model settings and the steps to compile the model code in a Unix-like computer (such as Linux or Mac). Chapter 4 "Inputs and outputs" describes the input data for model simulations and the output files and variables model simulations produce.

Please contact tanzeli1982@gmail.com if any errors in the user manual.

2. Scientific description

The major physical and biogeochemical processes that ALBM represents are shown in Figure 1. With the current structure, ALBM can represent different lake physical and biogeochemical states and fluxes: (1) thermal regime related (water temperature, sediment temperature, ice thickness, snow thickness, sensible heat flux, latent heat flux, upward shortwave radiation, upward longwave radiation and water-sediment heat conduction), (2) biogeochemistry related (dissolved OC, N_2, O_2, CO_2, CH_4 and P, gaseous N_2, O_2, CO_2 and CH_4, CO_2 diffusion, CH_4 diffusion and ebullition), and (3) ecosystem related (phytoplankton biomass and chlorophyll).

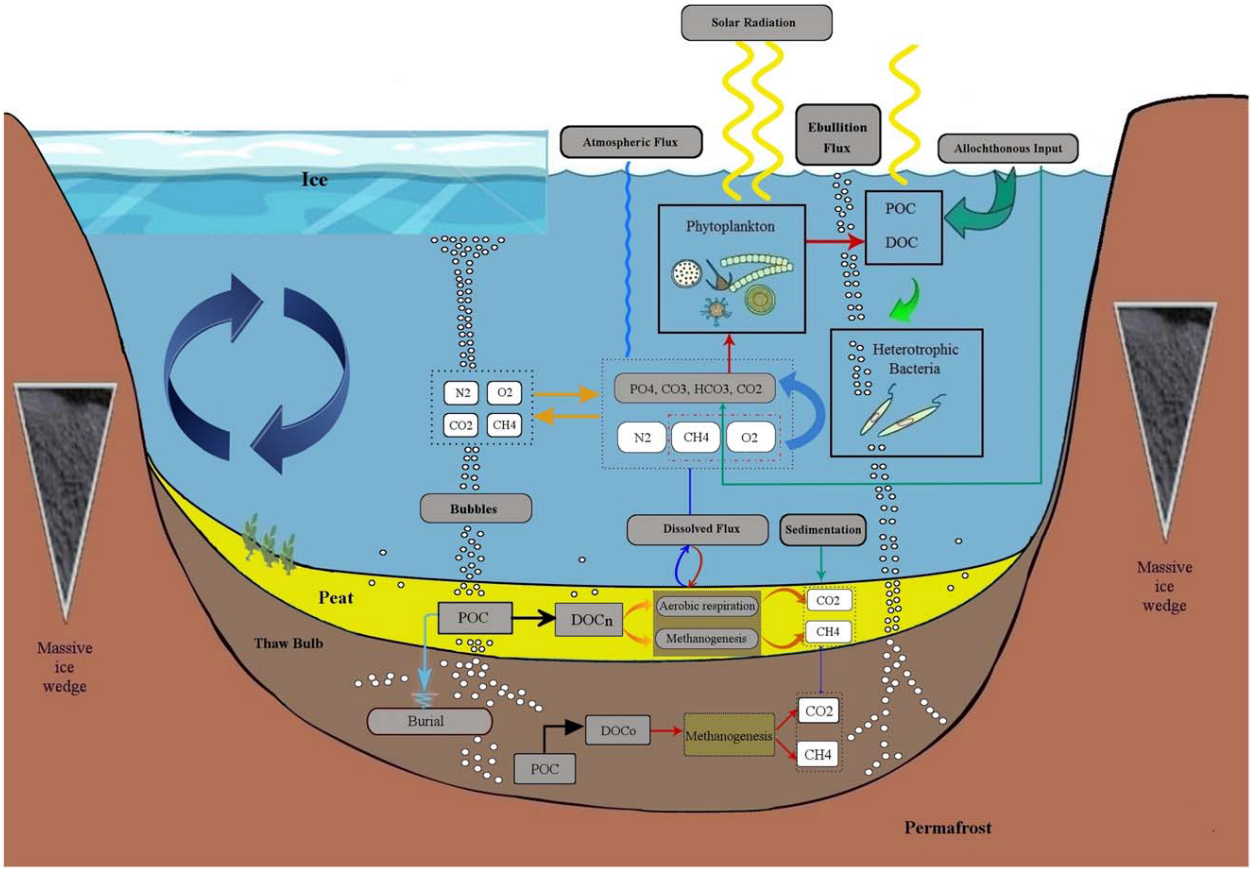

Figure 1. Simplified schematic of thermal, carbon and nutrient dynamics in ALBM.

2.1. Lake thermal dynamics

2.2. Radiative transfer

2.3. Lake carbon dynamics

3. Model setup

3.1. Code structure

The model consists of 26 Fortran code files that correspond to run control, I/O, utility functions, physical processes and biogeochemical processes. Table 1 lists these f90 files and their brief descriptions.

Table 1. Brief description of ALBM code files.

File name	Description
bayesian_mod.f90	subroutines for Monte Carlo based model calibration
bg_const_mod.f90	model constants for biogeochemical processes
bg_utilities_mod.f90	biogeochemical utility subroutines
bLake.f90	main file of the model
boundary_mod.f90	subroutines to process instant boundary conditions
bubble_mod.f90	subroutines to represent the transport of CH_4 bubbles
carbon_cycle_mod.f90	subroutines for carbon dynamics in the water column
costfunc_mod.f90	cost functions for model calibration
data_buffer_mod.f90	allocated memories for model state and other large variables and subroutines to initiate boundary condition variables
diagenesis_mod.f90	subroutines for carbon dynamics in the sediment column
io_utilities_mod.f90	utility subroutines for I/O
math_utilities_mod.f90	utility subroutines for mathematical algorithms
phy_const_mod.f90	model constants for physical processes
phy_utilities_mod.f90	physics utility subroutines
radiation_io_mod.f90	I/O subroutines for the radiative transfer module
radiation_mod.f90	the radiative transfer module
read_data_mod.f90	I/O subroutines
sensitivity_mod.f90	subroutines for parameter sensitivity analysis
shr_ctrl_mod.f90	variables for model settings
shr_kind_mod.f90	variable precision definitions
shr_param_mod.f90	model calibration parameters
shr_typedef_mod.f90	user-defined variable types
sim_coupler_mod.f90	coupler for different modules and run control subroutines
simulation_mod.f90	regular parallel simulation subroutine
soil_thermal_mod.f90	subroutines for sediment thermal dynamics
thermal_mod.f90	subroutines for water thermal dynamics and ice phenology

3.2. Model settings

ALBM run is controlled by a namelist file ***namelist.bLake*** in the source code directory. In the namelist file, model settings are classified into eight groups: **general** (lake configurations), **simulation** (switch flags of lake modules and simulation start and end dates), **resolution** (model spatial resolutions), **bayesian** (settings for model calibration and sensitivity analysis), **radiation** (settings for the radiative transfer module), **rundata** (input data), **archive** (module outputs), and **dbg** (model debug use). Table 2 lists all the model settings and their brief descriptions.

Table 2. Brief description of the ALBM settings.

Setting name	**Description**
general	
run_mode	"regular", "bayesian" or "sensitivity"
lake_file	the lake file
lakeid_file	the file mapping lake id to lake name
lake_range	selection of lake ids from the lake file for simulation
bthmtry_dir	the directory of lake bathymetry files
param_dir	the directory of lake-specific model parameter files
simulation	
Thermal_Module	switch flag for the thermal dynamics modules
Bubble_Module	switch flag for the CH_4 bubble module
Diagenesis_Module	switch flag for the sediment carbon dynamics module
Carbon_Module	switch flag for the water carbon dynamics module
Hydro_Module	switch flag for the lake hydrology module
Start_Year	simulation starting year
Start_Month	simulation starting month
Start_Day	simulation starting day
End_Year	simulation ending year
End_Month	simulation ending month
End_Day	simulation ending day
Spinup_Month	spin-up starting month

Spinup_Day	spin-up starting day
nSpinup	the spin-up period in years
resolution	
NWLAYER	the number of water layers
NSLAYER	the number of sediment layers
NRLAYER	the number of bubble radius bands
bayesian	
NMAXSAMPLE	the number of MC parameter samples
sample_range	selection of samples for model calibration or sensitivity simulations
obs_dir	the directory of lake observation files
obs_var	the observational variables for calibration
obs_weight	the weight of observations in the cost function
mc_file	the parameter sample file
sa_file	the cost function output file
radiation	
solar_dir	the directory of solar radiation parameter files
gas_dir	the directory of atmospheric gas optical characteristics files
albedo_dir	the directory of surface albedo characteristics file
co2_file	the atmospheric CO_2 concentration dynamics file
o3_file	the atmospheric O_3 concentration dynamics file
aod_file	the atmospheric optical depth (AOD) dynamics file
rundata	
forcing_tstep	the time step of atmospheric forcing data (“day” or “hour”)
forcing_dir	the directory of atmospheric forcing data files
hydro_dir	the directory of hydrological forcing data files
tref_file	the reference air temperature file
soc_file	the soil organic carbon file
veg_file	the tree coverage map file
wlnd_file	the wetland distribution map file

archive	
archive_tstep	the time step of model outputs (“day” or “hour”)
archive_dir	the directory to save model outputs
dbg	
DEBUG	switch flag to dump debug messages (for senior users only)
RESUBMIT	switch flag to resubmit the model simulation (for senior users only)

3.3. Build

Building the ALBM executable consists of three steps: 1) the installation or loading of requisite software, 2) the compilation of dependent packages, and 3) the compilation of the ALBM Fortran code.

First, ALBM needs a Fortran compiler (such as Intel Fortran or GNU Fortran) and a Message Passing Interface (MPI) package (such as OpenMPI or MVAPICH2) for compilation. On Linux clusters, the Fortran compiler and the MPI package are usually already installed. In such cases, they can be loaded by a command such as “***module load intel/19.0.3 mvapich2/2.3.1***”. On Linux or Mac PCs, the easiest way is to install such software through a package management tool (such as Homebrew for Mac and RPM for RedHat Linux).

Second, ALBM needs a parallel-netcdf package to handle the read and write of NetCDF files in the parallel mode. The source code of the parallel-netcdf package (**parallel-netcdf-1.4.0**) is already available in the ***libraries*** folder of the GitHub repository. The compilation of this package consists of five steps: 1) loading the Fortran compiler and the MPI package described above, 2) switching into the source directory of parallel-netcdf-1.4.0, 3) checking the pre-requisite environments for compilation by running the command “***./configure --prefix=install_directory***”, 4) compiling the code by running “***make***”, and 5) checking the compilation and copying the package libraries and executables into the specified installation directory at the 3rd step by running “***make install***”.

Third, once the first two steps are successful, the compilation of the ALBM Fortran code is straightforward. It consists of two steps: 1) modifying the make script ***make.sh*** in the source directory by pointing NETCDF_HOME to the parallel-netcdf installation directory and changing the loaded Fortran compiler and the MPI package, and 2) running the command “***./make.sh***”. The generated executable **ALBM.exe** will be located in the source directory.

3.4. Run

To run ALBM on a Linux cluster, there are two steps: 1) modifying the NETCDF_HOME variable and the loaded Fortran compiler and the MPI package in the job submission script ***bLakeJob.sub*** in the source directory (similar to the modification of ***make.sh***), and 2) submitting a job by running “***sbatch bLakeJob.sub***”.

To run ALBM on a Linux or Mac PC, there are also two steps: 1) copying the commands of setting NETCDF_HOME and LD_LIBRARY_PATH and loading packages from the job submission script ***bLakeJob.sub*** to the command terminal; and 2) running the model by “***mpirun -np #cores ./ALBM.exe namelist.bLake***”.

4. Inputs and outputs

4.1. Inputs

The atmospheric forcing data ALBM uses must be in a NetCDF-3 file. An example of the NetCDF-3 file structure is shown in Table 3. This file includes ten atmospheric forcing variables.

Table 3. The structure of the atmospheric forcing data file.

```
netcdf forcing_obs_Allequash {
dimensions:
        time = UNLIMITED ; // (13880 currently)
variables:
        int date ;
                date:long_name = "Date of the first data record" ;
                date:units = "YYYYMMDD" ;
        float tas(time) ;
                tas:long_name = "near-surface daily mean air temperature" ;
                tas:units = "K" ;
                tas:_FillValue = 1.e+20f ;
        float tasmin(time) ;
                tasmin:long_name = "near-surface daily minimum air temperature" ;
                tasmin:units = "K" ;
                tasmin:_FillValue = 1.e+20f ;
        float tasmax(time) ;
                tasmax:long_name = "near-surface daily maximum air temperature" ;
                tasmax:units = "K" ;
                tasmax:_FillValue = 1.e+20f ;
        float hurs(time) ;
                hurs:long_name = "relative humidity" ;
                hurs:units = "%" ;
                hurs:_FillValue = 1.e+20f ;
        float ps(time) ;
                ps:long_name = "surface air pressure" ;
                ps:units = "Pa" ;
                ps:_FillValue = 1.e+20f ;
        float pr(time) ;
                pr:long_name = "total precipitation" ;
                pr:units = "kg/m2/s" ;
                pr:_FillValue = 1.e+20f ;
        float prsn(time) ;
                prsn:long_name = "snowfall flux" ;
                prsn:units = "kg/m2/s" ;
                prsn:_FillValue = 1.e+20f ;
        float rsds(time) ;
                rsds:long_name = "surface downwelling shortwave radiation" ;
                rsds:units = "W/m2" ;
                rsds:_FillValue = 1.e+20f ;
        float rlds(time) ;
                rlds:long_name = "surface downwelling longwave radiation" ;
                rlds:units = "W/m2" ;
                rlds:_FillValue = 1.e+20f ;
        float sfcWind(time) ;
                sfcWind:long_name = "near surface wind speed" ;
                sfcWind:units = "m/s" ;
                sfcWind:_FillValue = 1.e+20f ;

// global attributes:
                :source = "Arctic Lake Biogeochemistry Model v2.0" ;
                :comment = "Observed meteo forcing data for Allequash_Lake Lake from 1/1/1979 to 12/31/2016" ;
}
```

The hydrology input file must also be a NetCDF-3 file. An example of the hydrology input file structure is shown in Table 4. The file includes eight hydrological variables.

Table 4. The structure of the hydrological input data file.

```
netcdf forcing_Harp_ice_CanESM2_hydro_rcp26 {
dimensions:
	time = UNLIMITED ; // (44409 currently)
variables:
	int time(time) ;
		time:long_name = "day" ;
		time:units = "days since 6/1/1978" ;
	float Qsi(time) ;
		Qsi:long_name = "inlet discharge" ;
		Qsi:units = "m3/s" ;
		Qsi:_FillValue = -999.f ;
	float tQsi(time) ;
		tQsi:long_name = "inlet flow temperature" ;
		tQsi:units = "celsius" ;
		tQsi:_FillValue = -999.f ;
	float dQsi(time) ;
		dQsi:long_name = "inlet flow density" ;
		dQsi:units = "kg/m3" ;
		dQsi:_FillValue = -999.f ;
	float DICQsi(time) ;
		DICQsi:long_name = "inlet flow DIC concentration" ;
		DICQsi:units = "uM" ;
		DICQsi:_FillValue = -999.f ;
	float DOCQsi(time) ;
		DOCQsi:long_name = "inlet flow DOC concentration" ;
		DOCQsi:units = "uM" ;
		DOCQsi:_FillValue = -999.f ;
	float POCQsi(time) ;
		POCQsi:long_name = "inlet flow POC concentration" ;
		POCQsi:units = "uM" ;
		POCQsi:_FillValue = -999.f ;
	float SRPQsi(time) ;
		SRPQsi:long_name = "inlet flow soluble reactive phosphorus" ;
		SRPQsi:units = "uM" ;
		SRPQsi:_FillValue = -999.f ;
	float Qso(time) ;
		Qso:long_name = "outlet discharge" ;
```

```
                Qso:units = "m3/s" ;
                Qso:_FillValue = -999.f ;

// global attributes:
                :history = "2017-9-18_1:12:8 UTC-05:00 by Zeli Tan" ;
                :comment = "Meteo forcing data for Harp Lake (Ontario, Canada), 6/1/1978 to 12/31/2099" ;
}
```

As described in Section 3.2, there are also other input data needed for model simulations. If the user needs support for these input data, please contact tanzeli1982@gmail.com.

4.2. Outputs

The ALBM simulation will produce several output NetCDF-3 files with the name in the format of ***bLakeOut.varname.date0_date1.nc*** in which varname is the name of an output variable (such as "watertemp"), date0 is the simulation starting date constructed from the settings of Start_Year, Start_Month and Start_Day (such as "19790101"), and date1 is the simulation ending date constructed from the settings of End_Year, End_Month and End_Day (such as "20060101"). There is a special output file ***bLakeOut.zw.nc*** that provides the information of lake water layers. Table 5 lists the variables that ALBM can produce and their brief description.

Table 5. Brief description of ALBM output variables.

Variable name	Description
thermal dynamics	
icethick	ice cover thickness
lakeheatf	downward net heat flux at the air-lake interface
latentheatf	latent heat flux at the air-lake interface
lwup	upward longwave radiation at the air-lake interface
momf	momentum energy flux at the air-lake interface
sedheatf	upward heat flux at the lake-sediment interface
sensheatf	sensible heat flux at the air-lake interface
snowthick	snow cover thickness
swup	upward shortwave radiation at the air-lake interface
watertemp	water temperature

carbon dynamics

aquatic ecosystem

5. References

1. Tan, Z., Zhuang, Q., & Walter Anthony, K. (2015). Modeling methane emissions from arctic lakes: Model development and site-level study. *Journal of Advances in Modeling Earth Systems*, *7*, 459-483.

2. Tan, Z., Zhuang, Q., Shurpali, N. J., Marushchak, M. E., Biasi, C., Eugster, W., & Walter Anthony, K. (2017). Modeling CO_2 emissions from Arctic lakes: Model development and site-level study. *Journal of Advances in Modeling Earth Systems*, *9*, 2190-2213.

3. Tan, Z., Yao, H., & Zhuang, Q. (2018). A small temperate lake in the 21st century: dynamics of water temperature, ice phenology, dissolved oxygen, and chlorophyll a. Water Resources Research, 54, 4681-4699.

4. Guseva, S., Bleninger, T., Jöhnk, K., Polli, B. A., Tan, Z., Thiery, W., ... & Stepanenko, V. (2020). Multimodel simulation of vertical gas transfer in a temperate lake. Hydrology and Earth System Sciences, 24, 697-715.

5. Guo, M., Zhuang, Q., Tan, Z., Shurpali, N., Juutinen, S., Kortelainen, P., & Martikainen, P. J. (2020a). Rising methane emissions from boreal lakes due to increasing ice-free days. Environmental Research Letters, 15, 064008.

6. Guo, M., Zhuang, Q., Yao, H., Golub, M., Leung, L. R., & Tan, Z. (2020b). Validation and sensitivity analysis of a 1-D lake model across global lakes. In review.