

# Simstrat

## 1D k-epsilon lake model

### User Manual

1. Introduction.....	2
2. Latest model changes up to version 3.1.....	2
3. Model set-up .....	3
3.1. Physical.....	3
3.2. Biogeochemical (AED) .....	5
4. Input files.....	5
4.1. Numerical .....	6
4.2. Physical.....	6
4.3. Biogeochemical (AED) .....	11
5. Output .....	12
6. Parameter estimation .....	13
6.1. Introduction.....	13
6.2. Set-up .....	14
6.3. Output .....	15

## 1. Introduction

Simstrat is a model for the physical simulation of water reservoirs, including basin morphology, interaction with the atmosphere, inflow and outflow.

A reservoir is simulated as a 1D vertical water column that is horizontally averaged. The column is composed of a certain number of layers, the evolution of which is driven by the atmospheric forcing, allowing the parameterization of stratification, energy transfers, turbulence effects, seiches, etc. Physically, water velocities, turbulent kinetic energy and its dissipation rate ( $k-\epsilon$ ), temperature, salinity, seiche energy, stress and buoyancy are modeled.

This document is a user manual. For an in-depth description of how the model works (governing equations, numerical schemes, parameterization, etc.), the reader is referred to the paper by Goudsmit, G-H. et al. (2002): "Application of  $k-\epsilon$  turbulence models to enclosed basins: The role of internal seiches." in *Journal of Geophysical Research: Oceans* (1978–2012) 107.C12: 23-1. Further changes to the physical model (i.e. which are not presented in this paper) are described in Chapter 2.

## 2. Latest model changes up to version 3.0

After the publication of the above-referenced paper, a few modifications have been performed on the algorithms governing the physical model (implemented in version 1.6):

- The model tended to over-estimate wind-induced vertical mixing in winter (in non-stratified conditions), and to underestimate it during the stratified season. This is not surprising as one-dimensional models cannot account for horizontal gyres as well as two- or three-dimensional ones, and may give more energy to seiches than what really occurs in non-stratified conditions, when a basin is very difficult to excite vertically. It is now possible to feed the model with a time-series of pre-filtered wind, which will only be used for allotting seiche energy differently (equation 19 in Goudsmit et al., 2002). For example, it has been found that reducing the wind when it is not sufficient to trigger seiches motion (the duration of the wind event is small when compared to oscillation period of the basin) helps towards better modeling of the thermocline seasonal behavior. This setting can be enabled in the parameter file.
- Improved parameterization of heat fluxes according to Schmid and Köster (2016)
- Implementation of gravity driven inflows: one can let the inflow sink through the layers of the reservoir based on its density, entraining water with it and stopping when neutral buoyancy is reached. This can be particularly important because a one-dimensional model will first distribute an inflow across entire horizontal layers before it spreads vertically, therefore an arbitrary estimate of the inflow location can lead to great inaccuracies in compounds distribution and water column structure. This setting can be enabled in the parameter file.
- Implementation of surface bound in-/outflows: if the inflows are not gravity driven, one can either define them at a fixed spot in the morphology (i.e. subaquatic groundwater inflow) or let them vary with the water level (i.e. surface in- and outflows). The outflow is always placed manually and can be surface-bound or not.
- Implementation of a ice/snow model (based on MyLake)

- Object-oriented Fortran 2003 architecture
- Possibility to start a simulation from a previously stored “snapshot” of the model. No need to run the whole model again from the very beginning for frequently updated simulations based on new meteo data for example.
- It is possible to let Simstrat compute water albedo according to the data of Grishchenko in Cogley (1979) instead of fixing a constant value.
- Two different values can be adopted for the seiching parameter  $\alpha$  depending on the season (i.e. depending on stability threshold).
- Coupling with the biogeochemical model “AED” of the University of Western Australia (<https://github.com/AquaticEcoDynamics/libaed-water>)

### 3. Model set-up

#### 3.1. Physical

The physical model is run via its executable file, and is governed by a parameter file. The name of the parameter file can be given as first argument when calling the model executable; if nothing is given (or for example if the model is run with a double-click), then `simstrat.par` is the default (this will be the name used in the rest of this manual). This file specifies all input files, output locations, model settings and parameters. Table 1 shows an explanation of this file which is in JSON format. The model parameters (last part of Table 1) are better described in the above-referenced paper.

JSON key	Description	Typical value
Input		
Initial conditions	Path to initial conditions file	
Grid	Path to grid file / vector of grid / grid resolution	
Morphology	Path to morphology file	
Forcing	Path to forcing file	
Absorption	Path to light attenuation file	
Inflow	Path to inflow file	
Outflow	Path to outflow file	
Inflow temperature	Path to temperature inflow file	
inflow Salinity	Path to salinity inflow file	
Output		
Path	Path result folder (is created if non-existent)	
OutputDepthReference	1: Lake bottom, 2: Lake water surface	
Depths	Path to file / vector of depths / output depth resolution	
Times	Path to file / vector of times / output time resolution	
All	True: all variables are written out, false: only selected variables in “Variables” are written	
Variables	Vector of variable names (if all=false)	
ModelConfig		

MaxLengthInputData	Maximum size of initial input data (initial conditions, morphology, grid...)	1000
CoupleAED	Biogeochemistry model (false = off, true = on)	
TurbulenceModel	1:k-ε, 2:M-Y	1
SplitSeicheParameter	True: use a_seiche if N2 exceeds "strat_sumr" and a_seiche_w otherwise; false: always use a_seiche	false
StabilityFunction	1:constant, 2:quasi-equilibrium	2
FluxCondition	0:Dirichlet condition, 1:no-flux	1
Forcing	1:Wind+Temp+SolRad, 2:Wind+Temp+SolRad+VapP, 3:Wind+Temp+SolRad+VapP+Cloud, 4:Wind+HeatFlux+SolRad 5:Wind+Temp+SolRad+VapP+Incoming_long_wave	3
UserDefinedWaterAlbedo	True: Albedo is calculated by Simstrat (Grishchenko tables); false: albedo has to be defined manually	true
UseFilteredWind	Use filtered wind to compute seiche energy (if "true", one more column is needed in forcing file)	false
SeicheNormalization	1:max N <sup>2</sup> , 2:integral	2
WindDragModel	1:lazy (constant), 2:ocean (increasing), 3:lake (Wüest and Lorke 2003)	3
InflowMode	0: no inflow; 1: manual inflow; 2: density-driven	2
PressureGradients	Horizontal pressure gradients (0: off, 1: Svensson 1978 ( <b>not working</b> ), 2:bottom friction)	0
IceModel	0: off, 1: on	1
SnowModel	0: off, 1: on (needs an additional column in the forcing file: precipitation)	1
AEDConfig		
AEDConfigFile	Path to AED config file (*.nml)	
PathAEDinitial	Path to folder with AED initial condition files	
PathAEDinflow	Path to folder with AED inflow files	
ParticleMobility	0: off, 1: on	0
BioshadeFeedback	0: off (light absorption is read from Simstrat file), 1: on (light absorption is computed in AED)	0
BackgroundExtinction	Only used if BioshadeFeedback is on	0.2
BenthicMode	0: Only bottom most layer interacts with sediment, 1: flancs also interact with sediment (but sediment concentrations are not simulated)	1
Simulation		
Timestep s	Simulation timestep in seconds	100
Start year	Starting year (used for Albedo calculation)	
Start d	Simulation start in days since 01.01. of starting year	
End d	End time in days	
DisplaySimulation	Display in terminal (0: off, 1:when data is saved, 2: at every iteration)	1
Continue from last snapshot	True: continue from snapshot file (if available); false: ignore snapshot files	0
ModelParameters		

lat	Latitude for Coriolis parameter [°]	47
p_air	Air pressure [mbar]	960
a_seiche	Fraction of wind energy to seiche energy [-]	0.01
a_seiche_w	Fraction of wind energy to seiche energy [-] in winter	0.001
strat_sumr	If maximum N2 is smaller than “strat_sumr”, then a_seiche_w is used instead of a_seiche	
q_nn	Fit parameter for distribution of seiche energy [-]	1.00
f_wind	Fraction of forcing wind to wind at 10m [-]	1.00
c10	Wind drag coefficient (a physical constant around 0.001 if wind drag model is 1; a calibration parameter around 1 if wind drag model is 2 or 3) [-]	0.001 / 1
cd	Bottom drag coefficient [-]	0.002
hgeo	Geothermal heat flux [W/m2]	0.10
p_sw_water	Fit parameter for absorption of sunlight [-]	1.00
p_lw	Fit parameter for absorption of IR radiation from sky [-]	1.00
p_windf	Fit parameter for convective and latent heat flux [-]	1.00
beta_sol	Fraction of short-wave radiation directly absorbed as heat by water [-]	0.30
wat_albedo	User defined water albedo [-] (only used if UserDefinedWaterAlbedo = true)	0.08
p_sw_ice	Fit parameter for albedo of ice, snow-ice and snow [-] (only used if IceModel on)	1
freez_temp	Freezing temperature of water [°C]	0.05
snow_temp	Temperature of snow [°C]	1

Table 1 – Simstrat configuration file (simstrat.par)

### 3.2. Biogeochemical (AED)

Information for this section is available on

<https://aquatic.science.uwa.edu.au/research/models/AED/index.html> and a default AED parameter file can be found in the tests folder in the Simstrat repository.

## 4. Input files

The input files are opened and read by the model while it is running. For all these files, the given depths must be within the limits set in the lake morphology (depth is zero at the surface and negative as it decreases downwards), while the given times must fall in the frame set by the simulation start and end time. In files where a series of values is required, depths have to decrease monotonously while times have to increase monotonously.

Throughout the simulation, the given values will be linearly interpolated (in depth and time) to obtain values at the coordinates needed by the model. If these coordinates are outside the given range, the value of the nearest neighbor is used. The model does not tolerate missing values. The files can have an arbitrary extension but must be text files.

## 4.1. Numerical

### Grid

The entry given to the json key “Input.Grid” can either be a string (path to a file), a vector containing the grid points (meaning the borders or faces of the grid layers) or a value specifying the total number of grid points. If a file path is given, the file can contain again either a vector of values (mostly used for variable grid spacing) or a number specifying the total number of grid points. If the grid points are specified, one needs to make sure to include the top and bottom values as defined in the morphology file otherwise an error occurs and the simulation aborts.

### Output depths

The Output.Depths key specifies at which depths the model results will be written. It can either be a string (path to a file), a vector containing the output depths in [m] or a value specifying output resolution in [m]. If a path to a file is given, this file can again contain either all output depths in [m] or an output resolution in [m]. The key Output.OutputDepthReference indicates whether the output depths should be interpreted as absolute height above sediment (“bottom”) or as depth below water level (“surface”). If the reference is “surface”, the depths are written as negative depths below water table. Conversely, if it is “bottom”, the depths are written as positive depths above lake sediment.

### Output times

The Output.Times key specifies at which times the model results will be written. It can either be a string (path to a file), a vector containing the output times in [days] or a value specifying output time resolution in timestep units (i.e. if 100 is given, the output is written every 100 timesteps). If a path to a file is given, this file can again contain either all output times in [days] or the resolution [timesteps].

## 4.2. Physical

### Morphology

The key Input.Morphology specifies the shape of the basin by giving its surface area (positive) at various depths. The values should cover at least the entire depth range of the reservoir: from the initial surface (0 m depth) to bottom (with ideally 0 m<sup>2</sup> surface area). During the simulation, water level will not be allowed to rise above the depth of the first given value which can be 0 or any positive number for which one knows the surface area.

The first line of the file is a header; the next lines are the depths in [m] in the first column and surface areas [m<sup>2</sup>] in the second column. An example of this file:

z [m]	Area [m2]
0	500000
-5	450000
-10	410000
-20	332000
-40	175000
-50	100000
-60	0

### Initial conditions

The Input.Initial conditions key specifies the state of the water column at simulation start time. Depth-dependent values for several variables can be given. Having initial conditions that are close to reality help the model to reach a physically consistent state faster. The depth values in the first column refer to the depth values in the morphology file. The first depth value is taken to be the initial water level of the reservoir (i.e. if -3 is chosen, the initial water level is set 3 meters below the 0 in the morphology file). The initial data values are extrapolated to the maximum depth in the morphology file in case not all the depths are given in the initial data file.

The first line of the file is a header, the next lines are the depths [m] in the first column and initial conditions in columns 2 to 7 (horizontal water velocity East U [m/s], horizontal water velocity North V [m/s], temperature T [°C], salinity S [‰], turbulent kinetic energy k [J/kg] and its dissipation rate  $\epsilon$  [W/kg]). An example of this file (with initial water table at 0) :

z [m]	U [m/s]	V [m/s]	T [°C]	S [‰]	k [J/kg]	eps [W/kg]
0	0.00	0.00	9.3	0.13	3e-06	5e-10
-5	0.00	0.00	9.2	0.13	3e-06	5e-10
-10	0.00	0.00	7.2	0.13	3e-06	5e-10
-15	0.00	0.00	5.2	0.13	3e-06	5e-10
-20	0.00	0.00	5.2	0.14	3e-06	5e-10
-40	0.00	0.00	5.1	0.14	3e-06	5e-10
-60	0.00	0.00	4.9	0.14	3e-06	5e-10

### Forcing

The forcing file specifies the atmospheric conditions to be applied at the reservoir surface throughout the simulation. At various times (in days), several parameters are specified, depending on the forcing mode chosen by the key ModelConfig.Forcing.

The first line of the file is a header; the next lines are the input: the structure of the columns is shown in Table 2.

Forcing mode	Column						
	1	2	3	4	5	6	7
<b>1</b>	Time [d]	Wind speed East [m/s]	Wind speed North [m/s]	Water surface temperature [°C]	Solar radiation [W/m <sup>2</sup> ]		
<b>2</b>	Time [d]	Wind speed East [m/s]	Wind speed North [m/s]	Air temperature [°C]	Solar radiation [W/m <sup>2</sup> ]	Vapor pressure [mbar]	
<b>3</b>	Time [d]	Wind speed East [m/s]	Wind speed North [m/s]	Air temperature [°C]	Solar radiation [W/m <sup>2</sup> ]	Vapor pressure [mbar]	Cloud cover [-]
<b>4</b>	Time [d]	Wind speed East [m/s]	Wind speed North [m/s]	Heat flux [W/m <sup>2</sup> ]	Solar radiation [W/m <sup>2</sup> ]		
<b>5</b>	Time [d]	Wind speed East [m/s]	Wind speed North [m/s]	Air temperature [°C]	Solar radiation [W/m <sup>2</sup> ]	Vapor pressure [mbar]	Incoming long wave radiation [W/m <sup>2</sup> ]

Table 2 – Structure for forcing file

If the use of filtered wind is enabled, one more column has to be added after the standard ones. It contains the filtered wind speed [m/s] (norm value). If the snow module is enabled (not necessary for ice!), precipitation data [m/h] has to be added at the end (only possible for forcing modes 2,3 and 5).

An example of this file (with forcing mode “3” and without filtered wind and precipitation):

t [d]	U [m/s]	V [m/s]	T [°C]	Sol [W/m <sup>2</sup> ]	Vap [mbar]	Cloud [-]
36556.0000	-0.87	-1.69	7.00	0.00	7.10	0.80
36556.0417	-0.98	2.41	7.20	0.00	7.10	0.46
36556.0833	3.80	-0.17	7.40	1.00	7.10	0.46
36556.1250	3.04	-2.90	7.50	0.00	7.10	0.46
36556.1667	5.20	2.99	7.40	0.00	7.10	0.40
36556.2083	3.47	1.99	6.90	0.00	7.10	0.65
36556.2500	-1.83	3.22	6.90	0.00	7.10	0.65
36556.2917	-0.91	3.79	7.00	0.00	7.00	0.55
36556.3333	-2.05	-2.84	7.30	26.00	7.00	0.20
36556.3750	4.76	1.16	8.20	189.00	6.80	0.10

## Light attenuation



The light absorption file specifies the attenuation coefficient of solar radiation as a function of depth and time. Here, the zero depth always represents the water surface (even if its absolute position varies during the simulation).

The first line of the file is a header, the second line gives the number of depths for which the attenuation coefficient is specified (say  $n$ ), the third line represents these depths (with the first number being a dummy value), the next lines are times [d] in the first column and attenuation coefficients [ $\text{m}^{-1}$ ] in columns 2 to  $n+1$ . An example of this file:

t (1.column)	z (1.row)		Abs [ $\text{m}^{-1}$ ]
2			
-1	0	-5	
0	0.200	0.300	
2130	0.212	0.331	
2260	0.177	0.198	
2390	0.667	0.668	
10000	0.700	0.750	

In this example, the light absorption coefficient on day 0 would be  $0.2 \text{ m}^{-1}$  at the surface, then linearly increase to  $0.3 \text{ m}^{-1}$  at 5 m depth, and remain constant below this depth.

### Inflow and outflow

Four files define the flows entering and coming out of the simulated reservoir, as a function of depth and time: water inflow, water outflow, temperature input and salinity input. Their contents represent a different physical quantity, but their structure is similar.

There are two kinds of different inflows: “deep inflows” and “surface inflows”. The inflow depth of “deep inflows” is fixed relative to the bottom of the lake (i.e. does not vary with varying lake level), while the inflow depth of “surface inflows” is fixed relative to the water surface. If the key `ModelConfig.InflowMode` is set to “1”, both types of inflows will enter the lake at the depths given in the file. If this key is set to “2”, then the “deep inflows” will plunge and stratify according to their density (surface inflows still enter the lake at fixed depths). In the following, the structure of the inflow files is given for both inflow modes.

- Manual inflow mode (`ModelConfig.InflowMode = 1`)

All values in the files must be given for a range of depths on a per-meter basis ( $\text{Q/h}$ ), as they will be integrated over depth by the model. Water inflow values must be positive, water outflow values must be negative. Temperature and salinity input can be either, as it can be used as an independent source or sink of T and S. In order to specify temperature (resp. salinity) of the inflowing water, the given values must be the product of the water inflow (as in the water inflow file) and the inflow temperature (resp. salinity), and thus be positive. In addition, the depths and times must match.

The first line of the file is a header, the second line gives the number of deep inflows (the ones that don’t move with the water level) and surface inflows (the ones that move with the water level). The third line represents these depths (with the first

number being a dummy value), the next lines are the times [d] in the first column, values (water inflow [ $\text{m}^2/\text{s}$ ], water outflow [ $\text{m}^2/\text{s}$ ], temperature input [ $^\circ\text{Cm}^2/\text{s}$ ] or salinity input [ $\text{‰m}^2/\text{s}$ ]) in columns 2 to  $nval+1$ .

Note that the depths are given relative to the initial water level (for deep inflows) and relative to the changing water level (for surface inflows).

- Density-driven inflow mode (ModelConfig.InflowMode = 2)  
For deep inflows, each column represents one density driven inflow with its input depth (from where it will move to its stratification depth) given in line 3 for inflow, temperature and salinity. For surface inflows and for the outflow (both deep and surface inflow), the manual syntax (see above) remains valid. From line 4 on, the actual inflows are given: times [d] in the first column, values (water inflow [ $\text{m}^3/\text{s}$ ], water outflow [ $\text{m}^2/\text{s}$ ], inflow temperature [ $^\circ\text{C}$ ] or inflow salinity [ $\text{‰}$ ]) in the second column.

An example of the water inflow file (left: manual inflow mode, right: density-driven inflow mode) for equal total inflow:

t (1.column)	z (1.row)		Q [ $\text{m}^2/\text{s}$ ]						t [d]	z (1. row)	Q [ $\text{m}^3/\text{s}$ ] / [ $\text{m}^2/\text{s}$ ]	
5	2								1	2		
-1	-10.0	-10.0	-5.0	-5.0	0	-2.0	0	-1	-10.0	-2.0	0	
3084	0.000	0.5	0.5	0	0	1	1	3084	2.5	1	1	
3098	0.000	0.55	0.55	0	0	1	1	3098	2.75	1	1	
3112	0.000	0.6	0.6	0	0	1	1	3112	3.0	1	1	

An example of the water outflow file with deep and surface outflow (for a neutral water balance with the inflow given above):

t (1.column)	z (1.row)		OutflowQ [ $\text{m}^2/\text{s}$ ]
0	2		
-1	-2.0	0	
3084	-2.25	-2.25	
3098	-2.375	-2.375	
3112	-2.5	-2.5	

An example of the temperature input file for a deep inflow at a temperature of 5°C and a surface inflow at 10°C with the inflow given above (left, manual) and for the case of a deep, density-driven inflows with temperatures of 5 and a surface inflow at 10°C (right, density-driven).

t (1.column)	z (1.row)		T [°C*m2/s]						t [d]	z (1.row)		T [°C] / [°C*m2/s]	
5	2								1	2			
-1	-10.0	-10.0	-5.0	-5.0	0	-2.0	0	-1	-10.0	-2.0	0		
3084	0.000	2.5	2.5	0	0	10	10	3084	5	10	10		
3098	0.000	2.75	2.75	0	0	10	10	3098	5	10	10		
3112	0.000	3.0	3.0	0	0	10	10	3112	5	10	10		

An example of the water inflow file (left: manual inflow mode, right: density-driven inflow mode), for an inflow at a salinity of 0.2‰ (both deep and surface inflows) with the inflow given above:

t (1.column)	z (1.row)		InflowS [‰*m2/s]						t [d]	z (1.row)		S [‰] / [‰*m2/s]	
5	2								1	2			
-1	-10.0	-10.0	-5.0	-5.0	0	-2.0	0	-1	-10	-2.0	0.0		
3084	0.000	0.10	0.10	0.0	0.0	0.2	0.2	3084	0.2	0.2	0.2		
3098	0.000	0.125	0.125	0.0	0.0	0.2	0.2	3098	0.2	0.2	0.2		
3112	0.000	0.15	0.15	0.0	0.0	0.2	0.2	3112	0.2	0.2	0.2		

If `ModelConfig.InflowMode` = 0, the calculation of vertical advection is deactivated. This is in particular useful in case inflows are negligible for the dynamics of the reservoir.

### 4.3. Biogeochemical (AED)

#### Initial conditions

In contrast to the initial conditions of the physical model (Simstrat), each biogeochemical variable needs its own file for the initial conditions. The reason for this is that AED modules can be turned on or off individually and thus the number of biogeochemical variables simulated can be highly variable. The names of the AED initial condition files are "AEDMODULE\_variable\_ini", for example the file for CH<sub>4</sub> of the AED carbon module would be "CAR\_ch4\_ini". These files need to be placed into the folder defined by `AEDConfig.PathAEDInitial`. The structure of the AED files is very similar to the Simstrat initial condition file:

z [m]	ch4 [mmol/m3]
0	0
-5	0
-10	0
-15	0
-20	5
-40	10
-60	30

### Inflow and outflow

The structure of AED and Simstrat inflow files is identical. For every simulated AED variable, there needs to be an inflow file named according to “AEDMODULE\_variable\_inflow” (for example for CH<sub>4</sub> of the carbon module: CAR\_ch4\_inflow) placed into the folder in AEDConfig.PathAEDInflow. For more details, the reader is referred to the “inflow and outflow” section for the physical model (4.2).

## 5. Output

The output is written to a separate text file for each output variable and stored in the location defined by Output.Path. If the output folder does not exist, it will be created automatically. The files are named according to the variable they contain var\_out.dat, where var is the short name of the variable (see Table 4). Output depth and times are used as defined in section 3.1. If Output.All is “true”, all variables in Table 4 are written out, if it is “false”, only the variables defined in Output.Variables are written out.

Short name	Description	Grid	Units
U	Water velocity (East direction)	Volume	m/s
V	Water velocity (North direction)	Volume	m/s
T	Temperature	Volume	°C
S	Salinity	Volume	‰
Qvert	Vertical advection	Face	m <sup>3</sup> /s
k	Turbulent kinetic energy	Face	J/kg
eps	Dissipation rate of turbulent kinetic energy	Face	W/kg
nuh	Turbulent diffusivity of temperature	Face	J·s/kg
num	Turbulent diffusivity of momentum	Face	m <sup>2</sup> /s
NN	Brunt-Väisälä frequency (stratification coefficient)	Face	s <sup>-2</sup>
B	Production rate of buoyancy	Face	W/kg
P	Production rate of shear stress	Face	W/kg
Ps	Production rate of seiche energy	Face	W/kg
HA	Long-wave radiation from sky	-	W/m <sup>2</sup>
HW	Long-wave radiation from water	-	W/m <sup>2</sup>
HK	Sensible heat flux	-	W/m <sup>2</sup>
HV	Latent heat flux	-	W/m <sup>2</sup>
Rad0	Solar radiation penetrating lake	-	W/m <sup>2</sup>
TotalIceH	Total ice thickness	-	m
BlackIceH	Black ice thickness	-	m
WhiteIceH	White ice thickness	-	m
SnowH	Snow height above ice	-	m
WaterH	Water depth (positive height above sediment)	-	m

Table 3 – Current Simstrat Output variables

## 6. Parameter estimation

### 6.1. Introduction

Parameter estimation is performed through the software package PEST<sup>1</sup>, which allows state-of-the-art model calibration and uncertainty analysis. More information about how the software works can be found in the PEST User Manual. In order to install PEST, one first has to download the archive containing all required files, and unzip it. The path to this directory must then be added to the PATH environment variable.

PEST requires several inputs that configure the parameter estimation for Simstrat:

- A control file (simstrat\_calib.pst) that specifies the parameter estimation setup: optimization settings, parameter values and ranges, field data, references to the other files, etc.
- A template file (simstrat\_par.tpl) that mimics the simstrat.par file, but with parameter names instead of values. Throughout the optimization process, PEST will fill it in with the values it wants and provide this new file to the model.

<sup>1</sup> <http://www.pesthomepage.org/>

- A batch file (simstrat.bat) which takes care of the execution of the model.
- Field data that can be used to calibrate the model (e.g. temperature profiles, salinity profiles, ice thickness measurements).
- Model results that can be directly compared to the field data (i.e. in the same units, at the same times and depths).
- An instruction file (simstrat\_obs.ins) that tells PEST how to relate the field data to the model results.

PEST can run in parallelized mode and operate much faster (using several CPUs, or several computers).

## 6.2. Set-up

The Python script PEST.py contains all the functions to prepare the whole PEST configuration and run calibration of Simstrat. As the single necessary input, it requires a configuration file (JSON format), which provides (i) the paths to the model, Simstrat configuration, observation files, and output directories/files, (ii) the number of CPUs to use in case of parallel calibration, (iii) the fixed and calibrated parameters.

The structure of the PEST configuration file is as follows:

JSON key	Description
files	
model	Simstrat executable
configFile	Simstrat parameter file
(obsFile_X)	Observation file(s) for variable X; the name X must correspond to the Simstrat output file X_out.dat. Examples: obsFile_T, obsFile_S, obsFile_IceH
refDate	Reference Simstrat date (corresponding to time=0), e.g.: "1981.01.01"
pestDir	Output directory for PEST to use as a working directory (PEST creates many files)
configFile_out	Output file to write the Simstrat parameter files with optimal parameters
results_out	Output directory to write the Simstrat results using optimal parameters
PEST	
nCPU	Number of CPUs to use (= 1 for single-threaded calibration, >1 for parallel calibration)
parameters	
(name of parameter as in Simstrat parameter file; max. 12 characters)	Fixed parameter(s): give a single value, e.g.: 1.0
(name of parameter as in Simstrat parameter file; max. 12 characters)	Calibration parameter(s): give a vector as [starting value, lower bound, upper bound], e.g.: [1.0, 0.5, 2.0]

An example of such a PEST configuration file is given on the GitHub: “Simstrat\_v2/testcase/TestCase\_LakeZurich.json”.

In order to launch parameter estimation, one has to run the function `runPEST(configFile)` from within Python, where “configFile” is the PEST configuration file. For our test case, the series of commands on a Windows command prompt would be as follows (assuming the working directory is at the location of both `PEST.py` and the PEST configuration file):

```
> python
> import PEST
> PEST.runPEST('TestCase_LakeZurich.json')
```

### **6.3. Output**

When running, PEST outputs several text files useful to understand (and possibly correct and improve) the calibration procedure. The set-up file (`simstrat_calib.pst`) contains the configuration used by PEST. The record file (`simstrat_calib.rec`) contains the whole log of calibration, for example the evolution of the performance as different parameter values are tested. If calibration succeeded, the parameter file (`simstrat_calib.par`) contains the final (calibrated) value of the parameters. The residuals file (`simstrat_calib.res`) contains the final residuals for all observations used for calibration. The run management record file (`simstrat_calib.rmr`) contains a log of the interactions between the processors (in the case of parallel calibration).