SimStrat
1D k-epsilon lake model

Developer Manual

Contents

eawag
aquatic research

Swiss Federal Institut of Aquatic Science and Technology, December 2016

I. Introduction

This developer manual is ...

II. Requirements

make
  Cmake
  GNU-Fortran compiler
  Alternatively Intel Fortran Compiler

III. Download and install SimStrat

IV. Code Structure

simstrat.f90

Main program. Calls Initialization, Form and simulation

subroutine **keps_simulation**	Simulation main loop, call to all the functions, coupling with FABM and management of the output
subroutine **Tridiagonal**	Tridiagonal matrix algorithm (solver)
subroutine **Coriolis**	Integration of the Coriolis force in the (x,y) velocities
subroutine **uvEquation**	Solving the advection-diffusion equations for the (x,y) velocities
subroutine **Buoyancy**	Calculation of the stratification coefficient
subroutine **Temperature**	Solving the temperature equation (transport, radiation, heat flux)
subroutine **TransportEquation**	Solving the compound equation (transport, source/sink)
subroutine **cmue_cn** and **cmue_qe**	Calculation of model parameters (cmue)
subroutine **Seiche**	Distributing the wind forcing into seiche energy
subroutine **Production**	Calculating shear stress and buoyancy energy production
subroutine **TKE**	Solving the turbulent kinetic energy (k) equation
subroutine **Dissipation**	Solving the TKE dissipation rate (Îţ) equation
subroutine **Advection**	Integration of the inflows and outflows to the model state
subroutine **Form**	Calculation of the morphologic parameters of the basin

simstrat_initialization.f90

contains subroutines to initialize the model

subroutine **Initialization**	Model initialization, calls to ParameterList, Morph and InitCond
subroutine **ParameterList**	Parameter initialization based on simstrat.par
subroutine **Morph**	Morphology initialization based on the morphology file

subroutine **Grid**	Vertical grid initialization based on the grid file
subroutine **InitCond**	Applying the initial conditions based on the initial conditions file
subroutine **InitCond_FABM**	Applying the biochemical initial conditions based on the corresponding files
subroutine **save_ini**	Preparation of the output grid based on the output depths and output times files
subroutine **check_advection**	Disabling of advection if all four inflow/outflow files are empty

---

keps_utilities.f90

contains subroutines that are used by other subroutines

subroutine **Interp**	Linear interpolation (nearest neighbour for values out of range)
subroutine **Interp_nan**	Linear interpolation (NaN for values out of range)
subroutine **Integrate**	Trapezoidal integration
subroutine **Forcing**	Calculation of all the relevant surface forcing parameters
subroutine **ReadForcing**	Reading the forcing file to obtain current values
subroutine **Absorption**	Reading the absorption file to obtain current values
subroutine **Lateral**	Reading the inflow/outflow files to obtain current values
subroutine **Lateral_FABM**	Reading the biochemical inflow/outflow files to obtain current values
subroutine **Lateral_rho**	Reading the inflow/outflow files to obtain current values, assuming that inflow placement is gravity-driven (not manual)
subroutine **Lateral_FABM_rho**	Reading the biochemical inflow/outflow files to obtain current values, assuming that inflow placement is gravity-driven (not manual)

V. Input Files

During a simulation, the model opens several files for reading and writing. Until it is closed, each open file is associated to a user-defined file ID (unit). Direct output to the screen is associated to unit 6 on most platforms. A list of the used files is given in Table 1.

Table 1: Input files used by SimStrat

File ID	File description	Reference	Use
10	Parameter	First command argument, or 'kepsilon.par' by default	Read once
11	Morphology	Parameter file, line 4	Read once
12	Grid	Parameter file, line 3	Read once
13	Initial conditions	Parameter file, line 2	Read once
14	Biochemical initial conditions (several files)	Path: parameter file, line 14	Read once
15	Output depths	Parameter file, line 8	Read once
15	Output times	Parameter file, line 9	Read once
20	Surface forcing	Parameter file, line 5	Read continuously
30	Light absorption	Parameter file, line 6	Read continuously
41	Water inflow	Parameter file, line 10	Read continuously
42	Water outflow	Parameter file, line 11	Read continuously
43	Temperature input	Parameter file, line 12	Read continuously
44	Salinity input	Parameter file, line 13	Read continuously
45-*	Biochemical input (several files)	Path: parameter file, line 14	Read continuously
80	Physical output (binary)	Parameter file, line 7	Write continuously
81-93	Physical output (text files)	Path: parameter file, line 7	Write continuously
101-*	Biochemical output (text files)	Path: parameter file, line 15	Write continuously

* means the number depends on which biochemical modules are enabled

VI. Discretisation Scheme and Numerical Approach

I. System of Governing Equations

The governing equations are described in Goudsmit et al. 2000

$$\frac{\partial T}{\partial t} = \frac{1}{A}\frac{\partial}{\partial z}\left(A(\nu'_t + \nu')\frac{\partial T}{\partial z}\right) + \frac{1}{\rho_0 c_p}\frac{\partial H_{sol}}{\partial z} + \frac{dA}{dz}\frac{H_{geo}}{A\rho_0 c_p} \tag{1}$$

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{1}{A}\frac{\partial}{\partial z}\left(A(\nu_t + \nu)\frac{\partial u}{\partial z}\right) + f v \\ \frac{\partial v}{\partial t} &= \frac{1}{A}\frac{\partial}{\partial z}\left(A(\nu_t + \nu)\frac{\partial v}{\partial z}\right) - f u \end{aligned} \tag{2}$$

$$\frac{\partial k}{\partial t} = \frac{1}{A}\frac{\partial}{\partial z}\left(A\nu_k\frac{\partial k}{\partial z}\right) + P + P_{seiche} + B - \epsilon \tag{3}$$

$$\frac{\partial \epsilon}{\partial t} = \frac{1}{A}\frac{\partial}{\partial z}\left(A\nu_\epsilon\frac{\partial \epsilon}{\partial z}\right) + \frac{k}{\epsilon}\left(c_{\epsilon 1}\left(P + P_{seiche}\right) + c_{\epsilon 3}B - c_{\epsilon 2}\epsilon\right) \tag{4}$$

II. Numerical Concept

In the above system of partial differential equations the temporal change of a quantity formally consists of a diffusive component D and a source/sink component S:

$$\frac{\partial \phi}{\partial t} = D(t,\phi) + S(t) \tag{5}$$

For the numerical integration an implicit Euler method is applied:

$$\frac{\phi^{n+1}-\phi^{n}}{dt} = D(t^{n+1},\phi^{n+1}) + S(t^{n}) \tag{6}$$

The resulting system of discretized equations can be written as a system of linear equations of the form:

$$R\phi^{n+1} = \phi^{n} + dt \cdot S^{n} \tag{7}$$

where ϕ represents any of the above quantities (T, u, v, k, ϵ), and R is a tridiagonal matrix of the form:

$$R = \begin{bmatrix} bu & au & 0 & \dots & 0 \\ cu & bu & au & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & au \\ 0 & \dots & 0 & cu & bu \end{bmatrix} \tag{8}$$

where $bu = (1 - au - cu)$. This system of linear equations can be solved very efficiently by the tridiagonal matrix algorithm.

III. Discretization Scheme: Finite Volume Approach on a Staggered Grid

The discretisation follows a finite volume approach. The water column is divided into nz volumes with height h_z, an area A_z at the top and A_{z-1} at the bottom. The volume centres have indices from 1 to nz (bottom to surface). The volume faces have indices from 0 to nz. The mean flow quantities (T, u, v) are placed at the centre of the volumes. Turbulent quantities (k, ϵ) are assigned to the volume faces (the interface between two volumes). Within a volume, the diffusion coefficients are assumed to be equal to the diffusion coefficient assigned to the top face of the volume.

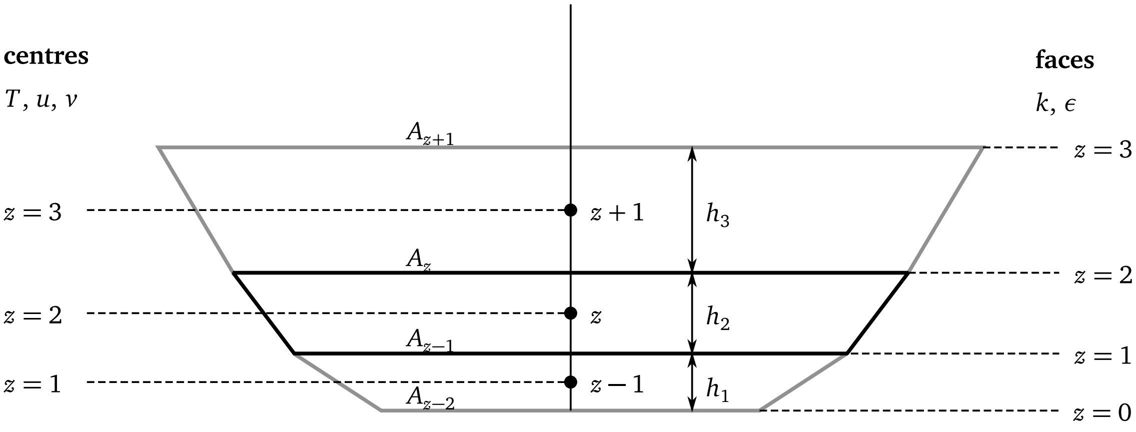

Figure 1: Discretisation Scheme

The temporal change of mean flow quantities within each volume results from the in-balance between the fluxes thought the two interfaces and sources/sinks:

$$\left.\frac{\partial \phi}{\partial t}\right|_{z} = \frac{A_{z-1}F_{z-1} - A_z F_z}{V_z} + S_z \tag{9}$$

where $V_z = h_z\,(A_z + A_{z-1})\,/2$.

The flux through the upper and lower interface are:

$$F_{face,z} = -\nu\,\frac{\partial\phi}{\partial z}\bigg|_{z}^{face} = -\nu_z\,\frac{\phi_{z+1}-\phi_z}{(h_{z+1}+h_z)/2} \tag{10}$$

$$F_{face,z-1} = -\nu\,\frac{\partial\phi}{\partial z}\bigg|_{z-1}^{face} = -\nu_z\,\frac{\phi_z-\phi_{z-1}}{(h_z+h_{z-1})/2} \tag{11}$$

The resulting discretization for mean flow quantities is then:

$$\frac{\phi_z^{n+1}-\phi_z^n}{dt} = \frac{1}{h_z\,(A_z+A_{z-1})\,/2}\left(A_z\,\nu_z^n\,\frac{\phi_{z+1}^{n+1}-\phi_z^{n+1}}{(h_{z+1}+h_z)/2} - A_{z-1}\,\nu_z^n\,\frac{\phi_z^{n+1}-\phi_{z-1}^{n+1}}{(h_z+h_{z-1})/2}\right) + S_z^n \tag{12}$$

This can be rearranged to:

$$\begin{aligned}
&dt\cdot form_1\cdot\nu_z^{n+1}\phi_{z-1}^{n+1}\\
&+(1-dt\cdot form_1\,\nu_z^n - dt\cdot form_2\,\nu_z^n)\phi_z^{n+1}\\
&+dt\cdot form_2\cdot\nu_z^{n+1}\phi_{z+1}^{n+1} = \phi_z^n + dt\cdot S_z^n
\end{aligned} \tag{13}$$

$$form_1 = \frac{-4A_{z-1}}{(A_z+A_{z-1})\,(h_z+h_{z-1})} \tag{14}$$

$$form_2 = \frac{-4A_z}{(A_z+A_{z-1})\,(h_{z+1}+h_z)} \tag{15}$$

The temporal change of turbulent quantities mainly follows the same concept except that the control volumes are shifted so that the turbulent quantities are at the center of the control volumes and the mean flow quantities are located at the volume faces. The balance for the turbulent quantities is therefore:

$$\frac{\partial\phi}{\partial t}\bigg|_z = \frac{(A_z+A_{z-1})\,/2\cdot F_z - (A_{z+1}+A_z)\,/2\cdot F_{z+1}}{V_z} + S_z \tag{16}$$

where $V_z = A_z\,(h_z + h_{z+1})\,/2$.

To calculate the flux through the upper and lower interface the turbulent kinetic energy and the dissipation at these interfaces are approximated by averaging the TKE and dissipation at the volume centres:

$$\nu_{z,face} = \left(\nu_{z,centre} + \nu_{z-1,centre}\right)/2 \tag{17}$$

$$\tag{18}$$

The flux through the upper and lower interface are:

$$F_{z+1} = -\nu\,\frac{\partial\phi}{\partial z}\bigg|_{z+1} = -\nu_{z,face}\,\frac{\phi_{z+1}-\phi_z}{h_{z+1}} \tag{19}$$

$$F_z = -\nu\,\frac{\partial\phi}{\partial z}\bigg|_{z} = -\nu_{z,face}\,\frac{\phi_z-\phi_{z-1}}{h_z} \tag{20}$$

The resulting discretization for turbulent quantities is then:

$$\begin{aligned}
&\frac{\phi_z^{n+1}-\phi_z^n}{dt} =\\
&\frac{2}{A_z\,(h_z+h_{z+1})}\left(\frac{A_{z+1}+A_z}{2}\,\nu_{z,face}^n\,\frac{\phi_{z+1}^{n+1}-\phi_z^{n+1}}{h_{z+1}} - \frac{A_z+A_{z-1}}{2}\,\nu_{z,face}^n\,\frac{\phi_z^{n+1}-\phi_{z-1}^{n+1}}{h_z}\right) + S_z^n
\end{aligned} \tag{21}$$

This can be rearranged to:

$$\begin{aligned}
&\left(dt\cdot form_{k1}\cdot v^{n+1}_{z,face}\right)\phi^{n+1}_{z-1}\\
+&\left(1-dt\cdot form_{k1}\,v^{n}_{z,face}-dt\cdot form_{k2}\,v^{n}_{z,face}\right)\phi^{n+1}_{z}\\
+&\left(dt\cdot form_{k2}\cdot v^{n+1}_{z,face}\right)\phi^{n+1}_{z+1}\\
=&\phi^{n}_{z}+dt\cdot S^{n}_{z} \qquad (22)
\end{aligned}$$

$$form_{k1}=\frac{A_z+A_{z-1}}{A_z\left(h_z+h_{z+1}\right)h_z} \qquad (23)$$

$$form_{k2}=\frac{A_{z+1}+A_z}{A_z\left(h_z+h_{z+1}\right)h_{z+1}} \qquad (24)$$

IV. Boundary Condition at the Sediment and Lake Surface

The basic discretisation assumes a no-flux boundary at the sediment and lake surface. Concentration dependent fluxes at these interfaces are implemented as an additional component of the main diagonal.