

MIKE ECO Lab

Short Scientific Description





PLEASE NOTE

COPYRIGHT

This document refers to proprietary computer software which is protected by copyright. All rights are reserved. Copying or other reproduction of this manual or the related programs is prohibited without prior written consent of DHI A/S (hereinafter referred to as "DHI"). For details please refer to your 'DHI Software Licence Agreement'.

LIMITED LIABILITY

The liability of DHI is limited as specified in your DHI Software Licence Agreement:

In no event shall DHI or its representatives (agents and suppliers) be liable for any damages whatsoever including, without limitation, special, indirect, incidental or consequential damages or damages for loss of business profits or savings, business interruption, loss of business information or other pecuniary loss arising in connection with the Agreement, e.g. out of Licensee's use of or the inability to use the Software, even if DHI has been advised of the possibility of such damages.

This limitation shall apply to claims of personal injury to the extent permitted by law. Some jurisdictions do not allow the exclusion or limitation of liability for consequential, special, indirect, incidental damages and, accordingly, some portions of these limitations may not apply.

Notwithstanding the above, DHI's total liability (whether in contract, tort, including negligence, or otherwise) under or in connection with the Agreement shall in aggregate during the term not exceed the lesser of EUR 10.000 or the fees paid by Licensee under the Agreement during the 12 months' period previous to the event giving rise to a claim.

Licensee acknowledge that the liability limitations and exclusions set out in the Agreement reflect the allocation of risk negotiated and agreed by the parties and that DHI would not enter into the Agreement without these limitations and exclusions on its liability. These limitations and exclusions will apply notwithstanding any failure of essential purpose of any limited remedy.





CONTENTS

1	Introduction	7
2	What Is Behind MIKE ECO Lab?	9
3	MIKE ECO Lab Set of Ordinary Differential Equations	11
3.1	Special handling of settling process	12
3.2	Special handling of light penetration in MIKE ECO Lab	13
3.3	Handling of built-in constants and forcings	14
3.4	Handling of site specific processes	14
3.5	Example of ordinary MIKE ECO Lab differential equation:	14
4	Integration With AD Engines	15
5	Integration Methods	17
5.1	Euler integration method	17
5.2	Runge Kutta 4th order	17
5.3	Runge Kutta 5th order with quality check	18





1 Introduction

MIKE ECO Lab is a piece of numerical simulation software for Ecological Modelling developed by DHI. It is an open and generic tool for customising aquatic ecosystem models to simulate for instance water quality, eutrophication, heavy metals and ecology.

MIKE ECO Lab functions as a module in the MIKE simulation software.

The module is developed to describe processes and interactions between chemical and ecosystem state variables. Also the physical process of sedimentation of state variables can be described (moves the state variable physically down the water column).

The module is coupled to the Advection-Dispersion Modules of the DHI hydrodynamic flow models, so that transport mechanisms based on advection-dispersion can be integrated in the MIKE ECO Lab simulation.

The description of the ecosystem state variables in MIKE ECO Lab is formulated as a set of ordinary coupled differential equations describing the rate of change for each state variable based on processes taking place in the ecosystem. All information about MIKE ECO Lab state variables, processes and their interaction are stored in a so-called generic MIKE ECO Lab template.



2 What Is Behind MIKE ECO Lab?

MIKE ECO Lab uses a so-called MIKE ECO Lab COM⁽¹⁾ object to perform the MIKE ECO Lab calculations. The MIKE ECO Lab object is generic and shared with a number of different DHI flow model systems. It consists of an interpreter that first translates the equation expressions in the MIKE ECO Lab template⁽²⁾ to lists of instructions that enables the object to evaluate all the expressions in the template. During simulation the model system integrates one time step by simulating the transport of advective state variables based on hydrodynamics. Initial concentrations or updated AD concentrations, coefficients/constants and updated forcing functions are loaded into the ECO Lab object and then the ECO Lab object evaluates all the expressions, integrates one time step, and returns updated concentration values to the general flow model system that advances one time step. An illustration of the data flow is shown in Figure 2.1.

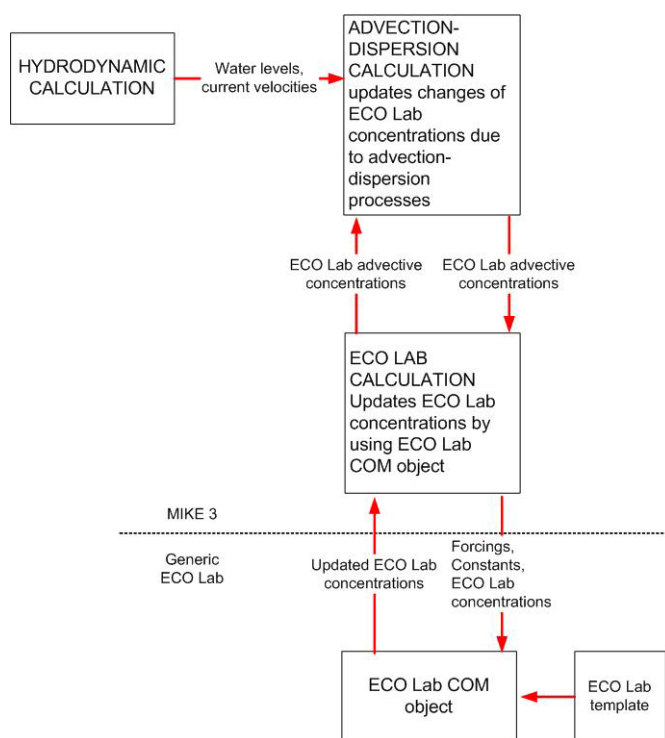


Figure 2.1 Data flow between the hydrodynamic flow model, in this case MIKE 3, and MIKE ECO Lab.

¹ Microsoft COM standard

² A MIKE ECO Lab template contains the mathematical definition of a MIKE ECO Lab model. It contains information about the included state variables, constants, forcings, processes and the state variables' rate-of-change differential equations.





3 MIKE ECO Lab Set of Ordinary Differential Equations

In general an ordinary differential equation is specified for each state variable.

The ordinary differential equation summaries the processes involved for the specific state variable. If a process affect more than one state variable, or the state variables affect each other, the differential equations are said to be coupled with each other.

The processes contain mathematical expressions using arguments such as numbers, constants, forcings and state variables. Processes always describe the rate at which something changes. In this context constants are values always constant in time, and forcings are values that can be varying in time.

$$P_c = \frac{dc}{dt} = \sum_{i=1}^n \text{process}_i \quad (3.1)$$

c: The concentration of the MIKE ECO Lab state variable

n: Number of processes involved for specific state variable

The Unit for 'Rate of change' of P_c can be specified as 3 types:

- g/m²/d
- mg/l/d
- Undefined

In general the part of the unit that relates to time shall always be specified as 'per day' in the template.

In MIKE ECO Lab there are two kinds of processes: transformation and settling processes. Transformation is a point description of a process not dependent on neighbouring points. Settling is a process transporting state variables to neighbouring points down the water column. The calculation of a state variable with a settling process is therefore dependent on information from neighbouring points. Also, the light forcing needs special handling to calculate the light penetration in the water column. A special built-in function can be used for this purpose. MIKE ECO Lab can also handle that some processes only take place at specific positions in the water column. For instance should reaeration (exchange with the atmosphere) only take place in the water surface. In other parts of the water column the reaeration is not active.

3.1 Special handling of settling process

The calculation of vertical movements need information from the layer below or above in multi layered systems. In MIKE ECO Lab, it is possible to specify one type of process with vertical movement: settling. This process is transporting the state variable vertically towards the bottom. As for transformation processes an expression must be specified describing the 'concentration change' from actual cell to cell below [mg/l/d]. When looking at a 'Settling' process directly in output from a MIKE ECO Lab simulation the output will show the result of the specified expression (the same as if it was a transformation process). The difference between a 'Settling' and a 'Transformation' process will appear in output of the affected state variable. This is because the numerical solution of a state variable affected by a 'Settling' process is different than if it was a 'Transformation' process. The definition of sign for a settling process is so that it should be specified as minus in the differential equation in order to transport the state variable correctly down the water column. The solution of a state variable with a settling process in a multilayered system takes into account that a contribution to the state variable is received from the layer above (if not top layer). Any variation in vertical discretization is also included in the numerical solution of differential equation involving a settling process. When solving a differential equation containing a settling process, MIKE ECO Lab substitutes the settling process expression in the differential equation with the following expression;

$$\frac{dc_n}{dt} = \frac{-settling_{n-1} \cdot dz_{n-1} + settling_n \cdot dz_n}{dz_n} \quad (3.2)$$

where

- $settling_{n-1}$ is the userspecified expression for 'rate of change' of the state variable concentration in layer n caused by a settling process transporting from layer n-1 to layer n [g·m-3·d-1]. It is usually a function of the concentration in layer n -1.
- $settling_n$ is the userspecified expression for 'rate of change' of the state variable concentration in layer n caused by a settling process transporting from layer n to layer n+1 [g·m-3·d-1]. It is usually a function of the concentration in layer n.
- dz_n is the thickness of layer n [m] and dz_{n-1} is the thickness of layer n-1 [m].

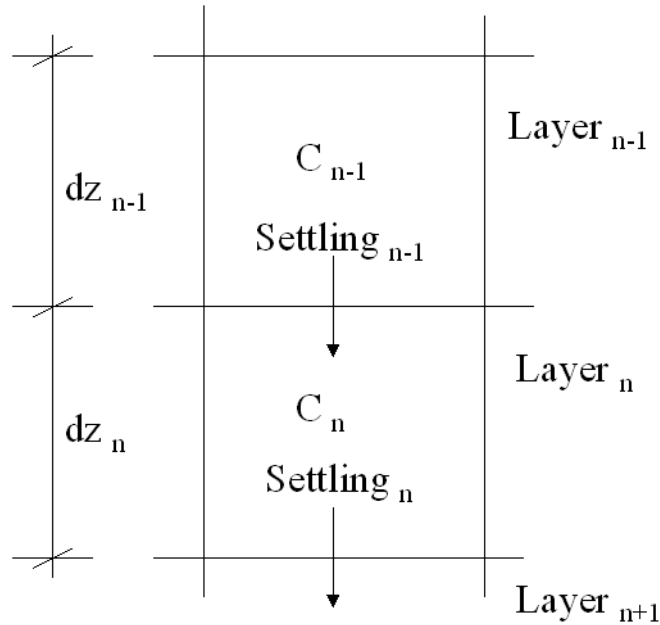


Figure 3.1 Schematic illustration of settling process

3.2 Special handling of light penetration in MIKE ECO Lab

Light penetration in the water column can be solved with a Lambert Beer built-in function in MIKE ECO Lab. In multi-layered systems with vertical varying extinction coefficients, the Lambert Beer expression must be calculated for each layer, and therefore, the Lambert Beer expression as argument uses the result of the Lambert Beer expression in the layer above.

$$I_n = I_{n-1} \cdot e^{\pi_n \cdot dz_n} \quad (3.3)$$

where I_n is the light available for primary production in the actual layer n , I_{n-1} is the irradiance in the layer above, π_n is the extinction coefficient and dz_n the layer thickness.

The way MIKE ECO Lab handles this problem is by using a so-called built-in function that is special designed to handle this 'Lambert Beer' problem. The functions are called:

- LAMBERT_BEER_1(surface radiation, layer height, light extinction coefficient). The function returns the solar radiation in top of each layer of the water column.
- LAMBERT_BEER_2(surface radiation, layer height, light extinction coefficient). The function returns the solar radiation in bottom of each layer of the water column.

3.3 Handling of built-in constants and forcings

Forcings such as for instance temperature can be specified in different ways. They can be user specified, as constant values or as timeseries, map series, or volume series. As an alternative to using user specified values of constants and forcings, it is also possible to use built-in forcing and constants. Built-in constants and forcings can be picked from a list in the dialog and they are already estimated in the hydrodynamic model, and they can be used as arguments in MIKE ECO Lab expressions. During simulation the built-in forcings and constants will be updated with the calculations in the hydrodynamic simulation. Examples of built-in forcings are temperature, flow velocities, salinity, wind velocity.

3.4 Handling of site specific processes

Some processes only take place in specific layers of the water column, and such processes are handled by calculating the process at the relevant layer where the process takes place and setting the process to zero in other layers. Examples of this could be a process such as re-aeration.

3.5 Example of ordinary MIKE ECO Lab differential equation:



Cyanide is assumed only to be affected by one temperature dependent decay process in this simple example;

$$\frac{dc_{cyanide}}{dt} = -decay$$

$$decay = K \cdot \Theta^{(temperature - 20)} \cdot c_{cyanide}$$

K: Decay coefficient (day⁻¹)

Θ: Arrhenius temperature coefficient

The scientific descriptions of specific MIKE ECO Lab differential equations and process equations in DHI supported MIKE ECO Lab templates has a PDF file attached containing scientific description of the template in question. For DHI projects with tailor-made MIKE ECO Lab templates for specific projects, the scientific description of the used MIKE ECO Lab equations typically will be described in the project report.



4 Integration With AD Engines

The dynamics of advective MIKE ECO Lab state variables can be expressed by a set of transport equations, which in non-conservative form can be written as:

$$\frac{\partial c}{\partial t} + u \frac{\partial c}{\partial x} + v \frac{\partial c}{\partial y} + w \frac{\partial c}{\partial z} = D_x \frac{\partial^2 c}{\partial x^2} + D_y \frac{\partial^2 c}{\partial y^2} + D_z \frac{\partial^2 c}{\partial z^2} + S_c + P_c \quad (4.1)$$

c : The concentration of the MIKE ECO Lab state variable

u, v, w : Flow velocity components

D_x, D_y, D_z : Dispersion coefficients

S_c : Sources and sinks

P_c : MIKE ECO Lab processes

The state variables may be coupled linearly or non-linearly to each other through the MIKE ECO Lab source term P_c

The transport equation can be rewritten as

$$\frac{\partial c}{\partial t} = AD_c + P_c \quad (4.2)$$

where the term AD_c represents the rate of change in concentration due to advection, dispersion (including sources and sinks).

The MIKE ECO Lab numerical equation solver makes an explicit time-integration of the above transport equations, when calculating the concentrations to the next time step.

An approximate solution is obtained in MIKE ECO Lab by treating the advection-dispersion term as AD_c as constant in each time step.

The coupled set of ordinary differential equations defined in MIKE ECO Lab are solved by integrating the rate of change due to both the MIKE ECO Lab processes themselves and the advection-dispersion processes.

$$c(t + \Delta t) = \int_t^{t + \Delta t} (P_c(t) + AD_c) + \partial t \quad (4.3)$$



The advection-dispersion contribution is approximated by

$$AD_c = \frac{c^* + (t + \Delta t) - c^n(t)}{\Delta t} \quad (4.4)$$

where the intermediate concentration c^* is found by transporting the MIKE ECO Lab state variable as a conservative substance over the time period Δt using the AD module.

The main advantage of this approach is that the explicit approach resolve coupling and non-linearity problems resulting from complex source MIKE ECO Lab terms P_c , and therefore the MIKE ECO Lab and the advection-dispersion part can be treated separately.

An implicit approach of solving the transport equations is not possible yet in MIKE ECO Lab.



5 Integration Methods

The following integration methods are available in MIKE ECO Lab: Euler, Runge Kutta 4, Runge Kutta with quality check.

5.1 Euler integration method

A very simple numerical solution method for solving ordinary differential equations.

The formula for the Euler method is:

$$y_{n+1} = y_n + h \cdot f(x_n, y_n) \quad (5.1)$$

which advances a solution y from x_n to $x_{n+1} = x_n + h$

5.2 Runge Kutta 4th order

A classical numerical solution method for solving ordinary differential equations. It has usually higher accuracy than the Euler method, but requires longer simulation times. The fourth order Runge-Kutta method requires four evaluations of the right hand side per time step.

$$y_{n+1} = \text{rk4}(y_n, f(x_n, y_n), x_n, h) \quad (5.2)$$

The function is solved this way:

$$k_1 = h \cdot f(x_n, y_n) \quad (5.3)$$

$$k_2 = h \cdot f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = h \cdot f\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = h \cdot f(x_n + h, y_n + k_3)$$

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} - O(h^5)$$

which advances a solution y from x_n to $x_{n+1} = x_n + h$

5.3 Runge Kutta 5th order with quality check

A numerical solution method for solving ordinary differential equations. The accuracy is evaluated and the time step is adjusted if results are not accurate enough.

$$y_{n+1} = f(y_n, f(y_n, x_n), x_n, h, \varepsilon, \text{yscale}) \quad (5.4)$$

The function is solved this way:

First take two half steps:

$$h_2 = 0.5 \cdot h \quad (5.5)$$

$$x_{n+1/2} = x_n + h_2 \quad (5.6)$$

$$y_2 = \text{rk4}(y_n, f(y_n, x_n), x_n, h_2) \quad (5.7)$$

$$y_2 = \text{rk4}(y_2, f(y_2, x_{n+1/2}), x_{n+1/2}, h_2) \quad (5.8)$$

Compare with one full time step:

$$y_1 = \text{rk4}(y_n, f(y_n, x_n), x_n, h) \quad (5.9)$$

Then estimate error:

$$y_1 = y_2 - y_1 \quad (5.10)$$

$$\text{err} = \text{MAX}(\text{ABS}(y_1/\text{yscale}))/\varepsilon \quad (5.11)$$

If the error is small ($\text{err} \leq 1.0$) the function returns

$$y_{n+1} = y_2 + \frac{y_1}{15} \quad (5.12)$$

which advances a solution y from x_n to $x_{n+1} = x_n + h$

or else the time step is reduced and the function tries again.



D	
data flow	9
I	
integration	17
L	
light penetration	13
P	
process	11
R	
re-aeration	14
S	
settling	12
T	
transport equation	15

