MIKE ECO Lab - 0D Engine

For testing your custom MIKE ECO Lab templates

Scientific Documentation & Examples

Powering **WATER DECISIONS**

MIKE 2021

DHI A/S
Agern Allé 5
DK-2970 Hørsholm
Denmark

+45 4516 9200 Telephone
+45 4516 9333 Support
+45 4516 9292 Telefax

MIKE@dhigroup.com
www.MIKEpoweredbydhi.com

Company Registration No.: DK36466871

PLEASE NOTE

COPYRIGHT

LIMITED LIABILITY

The liability of DHI is limited as specified in your DHI Software License 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

MIKE ECO Lab - 0D Engine
For testing your custom MIKE ECO Lab templates
Scientific Documentation & Examples

1 Introduction

Up to the release 2020 MIKE ECO Lab was only available as a module for various MIKE ECO Lab enabled hydrodynamic engines (MIKE 21, MIKE 3, MIKE SHE and M1D based engines). This meant that to test/run a MIKE ECO Lab template one needed to generate/update a proper hydrodynamic setup. With later releases a simplified hydrodynamic engine is available. This MIKE ECO Lab 0D engine provides a simple water column represented by exactly one rectangular grid element with one or more equidistant vertical layers and a specified depth. There is no water flow in or out this element. The MIKE ECO Lab 0D engine allows rapid calculation and testing of templates. The engine can also be used to run automatic calibration of some template parameters using the MIKE Autocal tool.

Templates with ABM classes are currently not supported by the MIKE ECO Lab 0D engine.

2 Running MIKEECOLab0D

The MIKE ECO Lab 0D engine can be run from the cmd line. The engine supports various parameter to specify the ECO Lab template, timing, in- and output options.

```
C:\>MIKEECOLab0D {parameter_1} {parameter_2}….
```

The computational speed is dependent on the template size and output definitions. As no HD/AD movement has to be computed the computational speed is rather high (usually some thousand timesteps per second).

2.1.1 Parameters

The MIKE ECO Lab 0D engine is controlled by a set of keywords and parameters. Keywords begin with a minus sign (“-“), directly followed by the keyword itself. Keywords have usually a short (one letter) or long form. Parameters following keywords must be separated by whitespace and can be either numeric or textual (string type). String types must not contain white spaces. Date/time parameters must be encompassed by double quotes.

Keyword / Parameter	Description
-f, -file {Filename} [experiment]	Read parameters set "experiment" from file
-t, -template {Filename}	MIKE ECO Lab template file name
-dx, -x [x-dimension]	X-dimension/length in m
-dy, -y [y-dimension]	Y-dimension/length in m
-d, -depth [depth]	Total water depth in m
-l, -layer [n-layer]	Number of equal spaced layers (1-n)
-n, -step [n-steps]	Number of time steps
-dt, -timestep [timestep sec]	Time step size in sec
-s, -start "[date]"	Simulation start date (year-month-day hour:minute:sec), must be enclosed by "
-e, -end "[date]"	Simulation start date (year-month-day hour:minute:sec), must be enclosed by "
-m, -method {solver}	Solver method (**Euler**, **RK4** or **RKQC**)
-v, -verbose +/-	Show detailed progress, lists all default outputs variables each step on console (default “-“)
-o, -output {symbol} {Filename} [freq]	Include output of "symbol" written to given file (dfs0 or csv/text). If symbol=”*” all default output items are written; if no filename given output will go to console

-i, -input {symbol} [value] [layer]	overwrite default value for "symbol" in layer [layer] with [value]; if no layer specified all layers will be affected
-i, -input {symbol} {Filename} [item] [layer]	Read input for "symbol" from "item"-th item in the file "Filename". Time varying inputs for state variables act as a source/sink term, i.e. the input value is added to the current value. Time varying input for other items will overwrite/replace the current value
-ft, -filetype {default extension}	specify default file extension (“.dfs0” or “.txt”)
-p, -precision {format}	Dfs0 data type, “float”,/“f” for single precision, “double”/”d” for double precision

(parameter format: numeric [i.i], textual {a_string})

Please note that string parameters should not contain any whitespaces (fx in file names) unless specified in a pfs parameter file.

Layer handling, Layer numbering

If multiple layers used the layer numbering follows the layer numbering in the MIKEFM engine series. This is from [bed] to [surface]:

```
Surface,      WS   layer [ n ]
Water column, WC   layer [n-1]
              WC   layer [...]
Bed           WB   layer [ 1 ]
```

Parameter files

Parameter files can be specified in two formats, either as simple text file or structured pfs file, supporting the same keywords as the cmd line interface.

Plain text files

MIKEECOLab0D supports simple, plain text files for parameter files. The parameter specification is the same as on the cmd line, i.e. instead of providing a set of parameters on the cmd line these can be saved to a simple text file. This format only supports one set of parameters belonging to the same experiment.

Example parameter text file

```
-t WQSimple.ecolab
-s “2019-09-01 00:00:00”
-e “2019-10-01 00:00:00”
-dt 1800
-method “RK4”
-v -
-o * Enterococci_Ecoli.ecolab.dfs0
```

Pfs format files

The other supported format is a parameter file in the pfs format. The pfs file format is a structured text file format used by most of DHI’s hydrodynamic engines. It is made of information blocks, so called pfs-sections and keyword-value pairs. A section start tag is defined by a section name inside squared brackets. The section end is indicated by a corresponding “EndSect” tag. Sections may be nested. Keyword-value pairs are specified by an assignment of one or more comma separated values to a key word (Keyword = value1, value2…). Keywords are standard identifiers, i.e. must start with a character and must not contain whitespace or extended characters. Numeric values can represent integer or float values. String values must be encompassed by a single quotes (“”).

If the pfs format is used specifications for multiple experiments can be saved in the same file. Each experiment is defined inside a [ECOLAB_EXPERIMENT] section. Parameters are specified as keywords with multiple values. The keywords correspond to the parameter identifiers on the cmd line. If multiple values are passed to the keywords they must be separated by commas. String parameters have to be encompassed by ' (i.e. "input = 'A','xyz.dfs0', 1, 2")

Example pfs section

```
…
[ECOLAB_EXPERIMENT]
  template  = 'WQsimple.ecolab'
  start = '2019-09-01 00:00:00'
  end = '2019-10-01 00:00:00'
  timestep = 1800 // half hour
  method = 'RK4'
  verbose = '-'
  output = '*', 'WQsimple.ecolab.dfs0'
EndSect
…
```

2.1.2 MIKE ECO Lab built-in constant and forcing

MIKE ECO Lab templates can make use of so called “built-in” constant and forcing. These represent input parameter that are directly coupled to the hydrodynamic engine and need no user input. The MIKE ECOLab 0D engine currently provides the following built-in constant/forcing:

Built-in ID	Description
AD_DT	AD timestep in sec
WQ_DT	WQ timestep in sec
TIMESTEP_PER_DAY	WQ timesteps per day
BED_AREA_OF_GRIDCELL	Bed/grid area
FLOODED_AREA_OF_GRIDCELL	Grid area
SURFACE_AREA_OF_GRIDCELL	Grid area
WATER_DEPTH	Total water depth
WATER_LAYER_HEIGTH	Current layer height
VOLUME_OF_GRIDCELL	Volume of grid cell
LAYER_CENTER_DEPTH	Depth of current layer centre

Default values for built-in constant/forcing

The numeric values for these built-in provided by the MIKE ECO Lab 0D engine are derived from the according parameter specification. If not overridden by user specification, the default values/parameters are:

-dx = 100 m
-dy = 100 m
-depth = 10.0 m
-layer = 1
-start = “2019-06-19 00:00:00”
-end = “2019-06-21 00:00:00”
-timestep = 300.0

This results in a standard cell area of 10000 m², a grid volume of 100000 m³, a standard depth and layer thickness of 10 m and a layer centre depth of 5 m. Standard AD and WQ timestep is 300 s and 288 timestep per day.

Experiment simulation period

The simulation period can be specified in various ways:

- If a start- and end-timestamp and the timestep size is given, the number of timesteps will be derived.
- If instead of the timestep size the number of timesteps are specified, the timestep size will be derived.
- If both timestep count and size are specified, the simulation end date will be derived.

2.1.3 User Inputs

Input can be provided for ECO Lab state variables, user constant and user forcing. The MIKE ECO Lab 0D engine allows time invariant and time varying input.

Time invariant inputs (constant over simulation time)

If the input value is constant over the simulation time and set at simulation start. The numeric value can be specified directly:

```
-input {symbol} [value] [layer]
```

Here “symbol” refers to an ECO Lab identifier in the template. If the symbol is not found in the template, an error message is printed. If no layer is specified, the input value is valid in all layers.

Time invariant inputs will overwrite the default values read from the template.

Time varying input

Input values can also be read from timeseries files (dfs0-format). The timeseries must cover the whole simulation period. If timestep sizes of the input file and the simulation experiment differ, linear interpolation will be applied. The general format to specify a time varying input is

```
-input {symbol} {filename} [file item] [layer]
```

If no layer is specified, the input value is valid in all layers.

Time varying inputs for state variables and other variables

Time varying inputs for state variables will act as source/sink terms. This means the input value will be **added** to the current state variable value. Input to ECO Lab constant or forcing items will set a new value for the item.

2.1.4 Outputs

The MIKE ECO Lab 0D engine can write simulation results as time series data. If no output filename is specified, the results will be printed to the console. If a filename is specified, the output will be stored in the specified timeseries as next item in the order of their specification. It is possible to specify an output frequency.

If a dfs0 filename is given, a MIKE dfs0 timeseries file will be created (existing files with the same name will be overwritten!). The default data format for dfs0 files is float/single precision; if needed, the format can be specified by the “-precision” keyword.

File extension other than “.dfs0” will result in a text output where data is stored as a comma separated value list, one line per timestep.

3 Examples

3.1.1 Basic example

In the following a simple example is shown to demonstrate the basic features of the MIKE ECO Lab 0D engine. It uses a very simple ECO Lab template that contains 4 state variables for the different ECO Lab scopes (water surface, water column, bed and sediment). Just the water column variable has a 1st order decay with a decay rate of k=0.693147 (~ -ln(0.5), thus one can expect a half time of one day). The template contains further all supported built-in constant and forcing plus a corresponding number of derived outputs to be able to observe their values. To demonstrate time varying input the template also has a spatial horizontal and vertical variable forcing called “Input” and an according auxiliary variable “InputTest” referring this forcing.

Basic simulation specifications

The template name is “Experiment1.ecolab” and it is loaded by the following keyword:

```
-t Experiment1.ecolab
```

The simulation is set to run 96 timesteps of 1800 s=30 min, so a total simulation period of 2 days. This corresponds to the following keywords:

```
-n 96
-dt 1800
```

The start date is not specified, so the default start date is used. The water column is set to be 10.0 x 10.0 x 10.0 m and to spawn over three layers. This corresponds to the following keywords:

```
-dx 10.0
-dy 10.0
-d 10.0
-l 3
```

Input handling

In the example the default values for some state variables will be overwritten. To demonstrate the setting of layer values the values for the variable “WC” will be defined as follows:

Surface (layer 3)	: WC = 10.0
Mid layer (layer 2)	: WC = 20.0
Bed (layer 1)	: WC = 30.0

This corresponds to the keywords

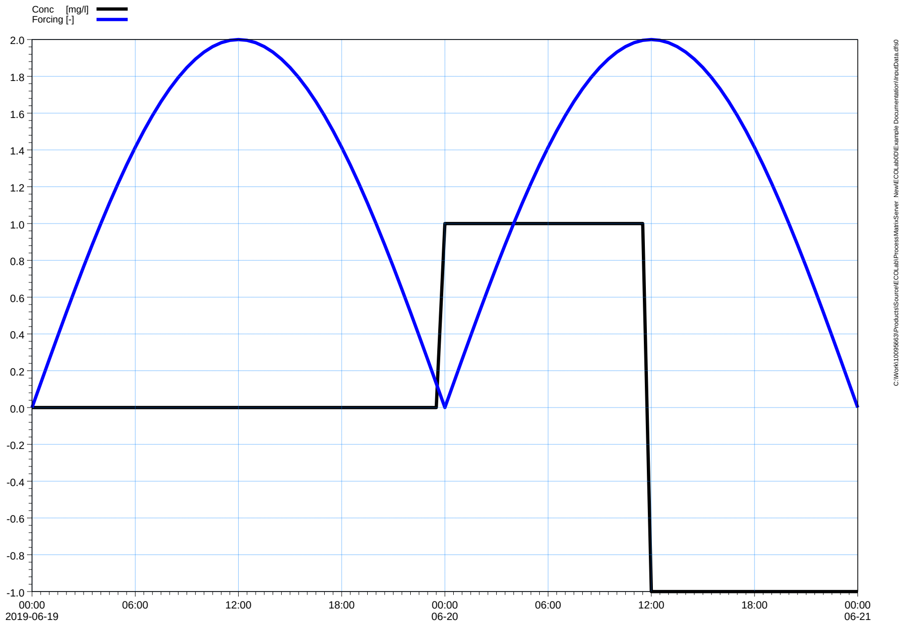
```
-i WC 10.0 3
-i WC 20.0 2
-i WC 30.0 1
```

Time varying input data

To demonstrate the handling of time varying input a timeseries file “InputData.dfs0” is included. This file has the same temporal coverage and resolution as the simulation. It contains two items. The first item represents a concentration variable. In the first half of the time period the item value is zero followed by a half day “+1” and “-1” the remaining period. The other variable type is undefined. Its value represents the positive part of a sine-function centred around 12:00:00, with an amplitude of 2 and a two-day period. It can be generated by the following equation in the time series editor:

```
i2 = abs(2*(sin((n-1)*acos(0)/24)))
```

A plot of the whole input series:



The first item of time varying input is applied to the state variable WS by the keyword

-i WS InputData.dfs0 1

The second item is used as input for the forcing “Input”. The timeseries data is used to update only the forcing in the 2nd layer by the keyword

-I Input InputData.dfs0 2 2

(here the 1st “2” represents the 2nd item of the time series and the 2nd “2” the layer number!)

Output handling

All state variables will be written to the same timeseries file. This corresponds to the keywords

-o WS Variables.dfs0
-o WC Variables.dfs0
-o BED Variables.dfs0
-o SED Variables.dfs0

To test the build-in constant and forcing the derived outputs are written to a separate result file. As ECO Lab derived outputs have no identifier one need to use the wildcard “*” to write all valid outputs with the following keyword

-o * AllVariables.dfs0

Parameter files

All the above listed keywords and parameters can be stored in a separate parameter file. In this example a simple text file “Experiment1.txt” is used where all defined keywords are stored, one keyword per line.

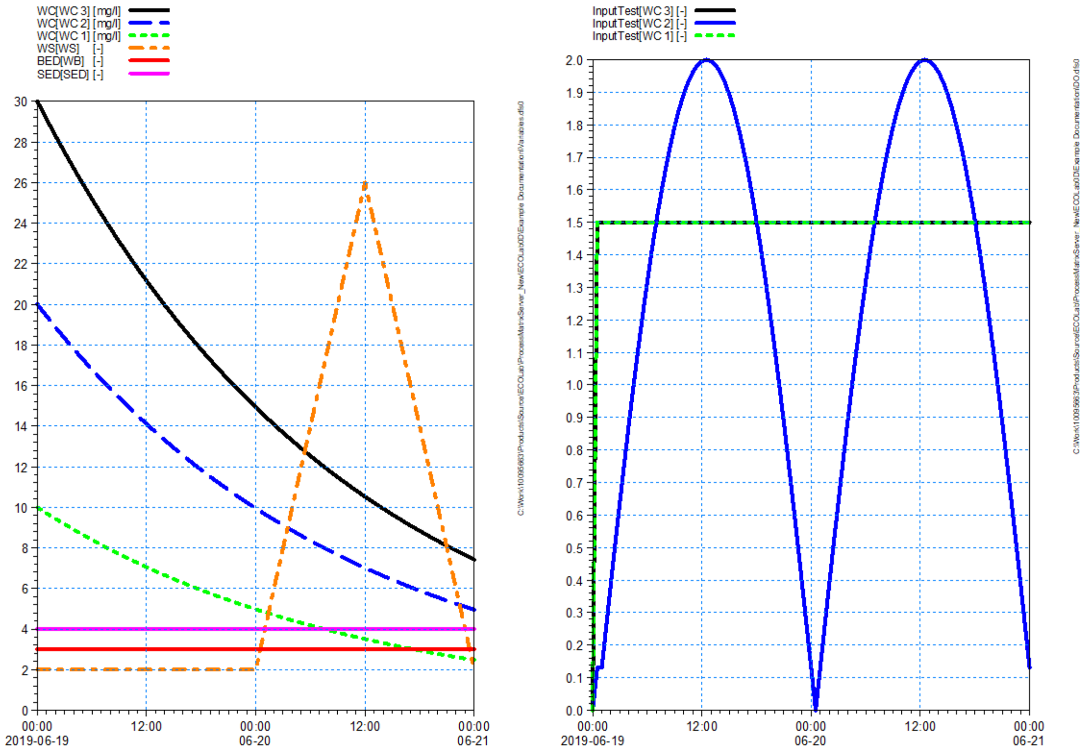
```
-t Experiment1.ecolab
-n 96
-dt 1800
-dx 10.0
-dy 10.0
-d 10.0
-l 3
-i WC 10.0 3
-i WC 20.0 2
-i WC 30.0 1
-i WS InputData.dfs0 1
-i Input InputData.dfs0 2 2
-o WS Variables.dfs0
-o WC Variables.dfs0
-o BED Variables.dfs0
-o SED Variables.dfs0
-o * AllVariables.dfs0
```

The experiment can be run by

```
C:>MikeECOLab0D -f Experiment1.txt
```

Results

The result plot “Experiment1.plc” will look like



Files provided:

Filename: Experiment1.ecolab
Description: MIKE ECO Lab template containing the test template

Filename: Experiment1.txt
Description: Experiment settings

Filename: InputData.dfs0
Description: Time series used for time varying inputs

Filename: Experiment1.plc
Description: Result plot

3.1.2 Prey-Predator ODF

In the following we use a simple Prey-Predator example in ECO Lab to demonstrate the use of the MIKE ECO Lab 0D engine. The template can be found in the template “PreyPredatorODF.ecolab”. The example is rather unusual for ECO Lab as it represents a classic, non-spatial set of Lotka-Volterra equations, coupling two populations. It is however well suited to demonstrate the use of the MIKE ECO Lab 0D engine and tools.

$$\frac{dPrey}{dt} = PreyGrowth - PreyMortality$$
$$\frac{dPredator}{dt} = PredatorGrowth - PredatorMortality$$

With the following process

PreyGrowth

$$PreyGrowth = r * \left(1 - \frac{Prey_0}{K}\right) * Prey_0$$

PreyMortality

$$PreyMortality = s * Predator_0 * Prey_0$$

PredatorGrowth

$$PredatorGrowth = v * Prey_0 * Predator_0$$

PredatorMortaliy

$$PredatorMortality = u * Predator_0$$

$Prey_0 = \max(0.0, Prey)$
$Predator_0 = \max(0.0, Predator)$

The following user constants are defined:

K	carrying capacity for prey biomass, default: 1.0
r	Prey growth rate, default: 1.3
s	Mortality effect of predator on prey biomass, default 0.5
v	Predator growth rate per prey biomass, default 1.6
u	Predator mortality, default 0.7

To run the model, execute the following cmd line:

```
c:\>MIKEECOLab0D -t PreyPredatorODF.ecolab -n 400 -dt 21600 -o Prey
test.dfs0 -o Predator test.dfs0
```

It will run the template “PreyPredatorODF.ecolab” with 400 timesteps (“-n 400”) and a timestep size of 21600 sec=6 hour (“-dt 21600”). The start date is not specified, so the default start date is used; the simulation end will be computed from the timestep count and size. It has two output definitions, one for the state variable “Prey” and one for “Predator”, both being saved into the same dfs0 timeseries “test.dfs0”.

The output on the cmd line will look like:

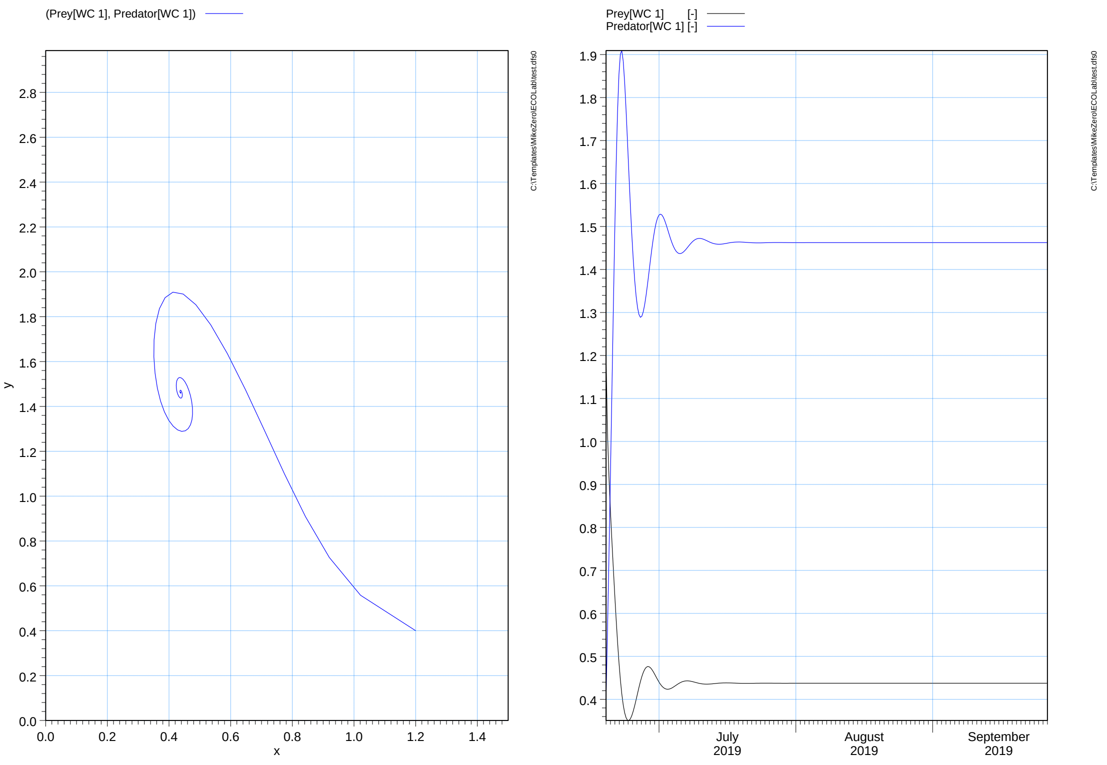
```
MIKE ECO Lab Experiment build Aug 23 2019, 12:59:35
Using ProcessMatrixServer V2119.03
Running ECO Lab Experiment on PreyPredatorODF.ecolab
Calculating simulation end from timestep size and count
New MIKE ECO Lab instance
Activating MIKE ECO Lab instance
============================= ECO Lab ===============================
Parsing ECO Lab/Oilspill/Particle templat
Using new parse
Template revision :
Optimising MIKE ECO Lab expressions enabled
Trigonometric angle unit is DEGREE
pH Options: ConstantSet=4, Sulfate=1, Borate=1, pH scale=
no spatial search tree used
Optimisations:
Parsing don
============================= ECO Lab ===============================
Initial conditions
Prey: 1.2
Predator: 0.4
K: 1
r: 1.3
s: 0.5
u: 0.7
v: 1.6
Start date : 2019-06-19 00:00:00
End date   : 2019-09-27 00:00:00
Running 400 timestep of 21600 sec
Output : test.dfs0, 2019-06-19 00:00:00 - 2019-06-19 00:00:00,
21600.000000, 400, Prey, Predator

2019-06-19 00:00:00  0 %
2019-06-24 00:00:00  5 %
2019-06-29 00:00:00  10 %
2019-07-04 00:00:00  15 %
2019-07-09 00:00:00  20 %
2019-07-14 00:00:00  25 %
2019-07-19 00:00:00  30 %
2019-07-24 00:00:00  35 %
2019-07-29 00:00:00  40 %
2019-08-03 00:00:00  45 %
2019-08-08 00:00:00  50 %
2019-08-13 00:00:00  55 %
2019-08-18 00:00:00  60 %
2019-08-23 00:00:00  65 %
2019-08-28 00:00:00  70 %
2019-09-02 00:00:00  75 %
2019-09-07 00:00:00  80 %
2019-09-12 00:00:00  85 %
2019-09-17 00:00:00  90 %
2019-09-22 00:00:00  95 %
```

```
2019-09-27 00:00:00  100 %
Run successful after 0.0925 sec (4347.83 ts/sec)
```

In the very top some general information on the used MIKE ECO Lab 0D engine are shown.

Then the used template and timing information is listed. The next block shows the initialisation messages from MIKE ECO Lab, followed by a list of initial values applied by the MIKE ECO Lab 0D engine and a summary of the defined output. During run the engine will print further outputs to the console. If no timeseries outputs are defined, a progress status message will be shown every 5% of the simulation time. The last line gives a status summary.

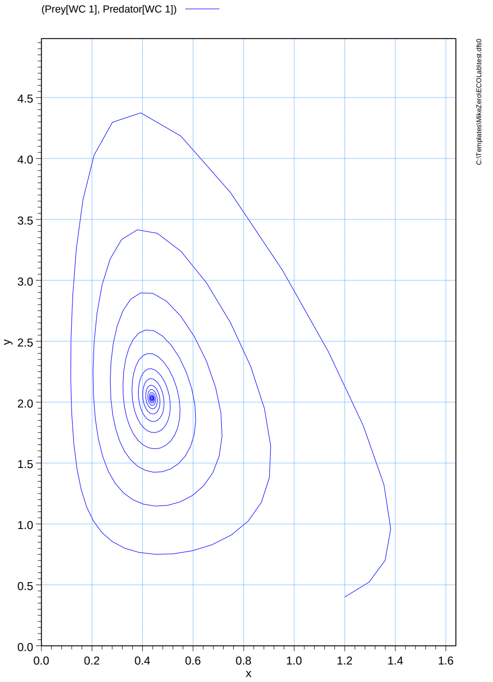
The following image shows a X/Y plot (a so-called phase plot) and a time series plot of the two state variables. The plots are stored in the plot composer setup “PreyPredatorODF.plc”. As one can see, the ODF reaches a stable distribution after few days.

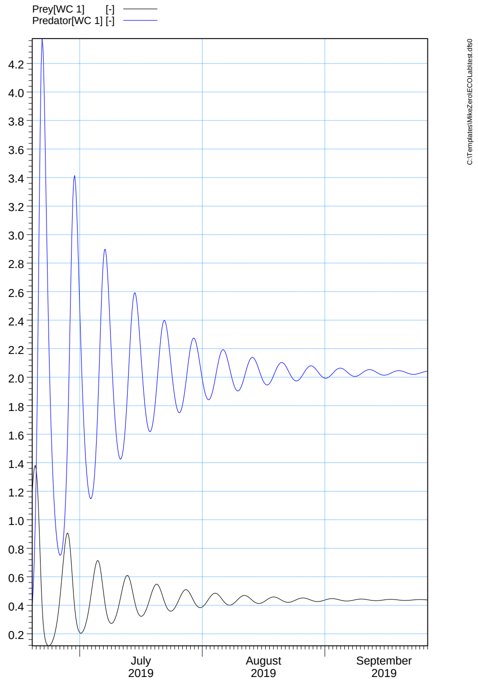


To change some input values, the cmd line can be modified. To specify a new value for the carrying capacity K = 2 use:

```
c:\>MIKEECOLab0D -t PreyPredatorODF.ecolab -n 400 -dt 21600 -i K 2 -o
Prey test.dfs0 -o Predator test.dfs0
```

If you have the result viewer still open press “F5” to update the data. Possibly you need to activate each plot to trigger a redraw. Otherwise just re-open the plot composer file. You may need to adjust the axis scaling to fit the timeseries data. This should result in the following plot.





With K=2 the prey-predator system shows a much longer damped oscillation. Feel free to experiment with the input parameters. Try for example K=3.289.

Files provided:

Filename: PreyPredatorODF.ecolab
Description: MIKE ECO Lab template containing the Lotka-Volterra equations

Filename: PreyPredatorODF.plc
Description: A plot composer setup with a phase plot and a timeseries plot of the result file “test.dfs0”