Lake model

WaSiM contains a complete approach to model natural and artificial lakes as an integral part of both, the unsaturated zone- or Richards-model and the routing model with detailed feedback from one model component to the other. Here, the functional design of the coupling between Richards model, routing model and the other affected models (irrigation, snow, evaporation, interception, groundwater etc.) is described.

Also, a time variable abstraction mode for abstractions from rivers and lakes and reservoirs and for lake outflow and reservoir outflow is implemented – internal abstractions may even take into account the capacity of the target channel to limit the abstraction value. Abstractions may now also be read in from an external file and they can be defined in a time variant way (diurnal and weekly courses can be defined).

General mechanisms of the lake model

The lake model is based on three mechanisms:

1. The POND-modelling as already implemented in the Richards-Version of WaSiM
2. The coupling between the routing model and the unsaturated zone model in order to balance the lake volume affected by inflows, outflows, and abstractions between the POND-cells of a lake.
3. A temporally high resolution balancing of all inputs and outputs for each time step – with statistic output for each lake.

The crucial point is, that a conventional POND-cell is modelled as a single cell without connection to the surrounding cells, whereas a lake by its nature must be able to balance the water table across all cells of the lake. Since the POND-modelling was already implemented in WaSiM, the main part of the lake model implementation was the balancing of the water volume between all lake-cells considering also different elevations of the lakes ground. Thus, a changing water level may change the area of the lake as well.

For a cell to be modelled as a lake cell, only the following few prerequisites are required:

- A grid containing codes for each lakes; equal codes mark cells for distinct lakes, e.g. all cells with code 1 belong to lake 1, all cells with code 2 belong to lake 2 etc Cells which are not part of a lake will have code -9999 (nodata). This grid describes the maximum extent of the lakes when the lakes are filled with the maximum possible amount of water.
- A grid containing the maximum depth of the lake for each grid cell in meters. This grid corresponds to the cells describing the lake cells (see topic above). This grid is the well known POND-grid already known to WaSiM-users from the unsaturated zone model (Richards-approach).
- A new paragraph in the control file containing some new parameters for lake modelling and some default evaporation parameters to be used by the potential-evaporation-algorithm
- considering the lakes in the routing descriptions (no change to existing control files required)

So it's very simple to set up lake modelling using a physical coupling between unsaturated zone, groundwater, evaporation and routing.

Note: The WaSiM-Topmodel-Version will NOT support the extended lake model. Only the traditional model entirely within the routing model is supported. Even if a POND-Grid is defined in the Topmodel-version, no precipitation and evaporation handling is done for the POND-cells.

Impact of the lake model on the other sub models

Evaporation model

For a lake cell, the potential evapotranspiration will be calculated depending on the water centent of the actual POND-cell:

If there is water in the POND-cell, then some default-parameters are used from the control file section "[lake_model]":

- Albedo (default 0.15)
- roughness length z0 (default 0.1)

other parameters will not be used in the evaporation scheme for lakes(and will hence not be available as a parameter in the lake_model section of the control file):

- surface resistance rsc (0 is assumed in the relevant relations)
- leaf area index LAI (net radiation is set to incoming global radiation * (1-albedo))
- Vegetation covered fraction vcf (assumed to be 1.0)

Using these parameters, the Penman-Monteith evaporation scheme is used like for all other land use codes.

If there is NO water in the POND-cells, then the land use code from the land use grid is used and the evaporation will be calculated in the traditional manner.

When multilayer land use is enabled, then only the lowest land use layer will be handled as lake. The other land use layers will still be handled as normal land use layers. This scheme takes into account the flooding of low lands, where the bushes and trees may still be able to transpirate in the traditional way.

For a "real" lake, meaning a lake, which has a more ore less stationary perimeter, the lake cells may be represented by a bare soil in the land use grid, resulting in fast drying conditions when the water level drops. The cells beneath the lake cells may be represented by their normal land use, e.g. Forest, bushes and grass. When the lake water level rises, then trees and bushes will still transpire, but the flooded grassland will then be handled as open water.

Furthermore, if the cell is a valid LAKE cell with water in it, then no potential interception is calculated for the lowest layer and no potential evaporation from the soil will be calculated (each set to 0).

Interception model

The scheme for calculating interception evaporation remains the same like in WaSiM 7.10.x. Since the potential interception evaporation was already set to zero in the evaporation model (see above), there was no change required.

Snow model

Snow accumulation will be supressed by setting the snow fraction on the precipitation to zero for POND cells which are also lake-cells and which contain any water. There is now estimation of freezing (e.g. no ice coverage) – so the snow falling into a lake will melt immediately.

Irrigation model

There is no change to the irrigation model, either. However, during initialization, all cells which have an irrigation code > 0 are set to code 0, so no irrigation will be calculated for such cells at all.

Unsaturated Zone (Richards) model

When calculating the water balance for the soil column, the cells for a lake are handled very similar to the handling of the normal POND-cells: The water within the lake is represented by a pond. So all algorithms applied to the unsaturated zone model will be applied to the lake cells, too.

However, there are some initialization issues for lake cells:

b) Drainage is switched off (by setting the drain level to -1)

c) Interflow is switched off (by setting the slope to 0)

d) Irrigation is switched off (by setting irrigation codes to 0)

e) All soil layers are initialized as saturated and the groundwater level is set to soil surface (in order to have no unsaturated zone below a lake)

f) Depending on the initialization method set in the control file, all elevation values for the lake cells are artificially lowered by the value given in the MAXPOND-grid (which is the maximum lake depth for each grid cell). Alternatively, the original DEM may already contain the lake ground as elevation and not the lake surface, so this initialization may also be skipped.

g) All exfiltration and infiltration into and from rivers are switched off, as long as the cell has any water in the POND grid. However, if the cells falls dry, exfiltration and infiltration may still be calculated.

h) Macro pore infiltration is switched off, as long as the cell has any water in its POND-grid. If the cell falls dry, macro pore infiltration may occur again.

i) "Using ponds" must be activated in the control file in the unsaturated zone model

Groundwater model

The groundwater model is mostly not affected by the lake model: because the DEM-values are already lowered within unsatzon-initialization, the only parameter to be checked is the usage of POND-water for calculating the hydraulic heads. Since this option was already introduced in WaSiM 7.9.7 (November 2007), the actual change for lake modelling was to make sure, this parameter is really set to 1.

Thus, the groundwater model is always aware of the water within the lake, so changes in the lake water level may result in changes in the groundwater flow, too. The statistics written in the routing model contains the amount of water which flows between the groundwater and the lake.

Routing model

Here, the main changes were implemented. Until now, the routing model was not aware of the grid structure of the modelled basin. The routing was an entirely graph oriented algorithm. Even lakes and reservoirs were modelled as part of the directed graph only – no real localisation or mapping to distinct grid cells could be made.

The only feedback to other submodels was implemented by the so called "river flow table" containing the amount of water in each flow channel. These data were used in the irrigation model as well as for the infiltration of river water into groundwater and exfiltration from groundwater into rivers (base flow generation).

The new routing model is now aware of the grid structure of the model domain. To implement the much more complex water balance and abstraction handling, a re-factoring of the C++ class structure was required. Reservoirs and abstractions were encapsulated from the routing-class into separate classes, all methods for reading and writing input and output files were re-factored as instances of a new class FileBuffer-class.

How the lake model works

During initialization the following steps are carried out:

- for each lake, the range of rows and columns enclosing the lake is extracted from the lake grid. A lake is defined by a set of cells with a unique ID (integers starting from 1). So all cells in the lake grid with value 1 constitute the Lake with ID 1. Cells with value 2 constitute Lake 2 etc
- For each cell of the same lake, the values of the DEM (as already adjusted in the Richards model initialization to the MAXPOND-value) plus the MAXPOND-Value (the lakes depth for each cell) must result in the same altitude. Otherwise, the model will stop with an error message.
- Note: the MAXPOND grid must contain the **maximum** lake depth for each cell, see later in this document about the requirements of input grids.
- A lake is allowed to be located in one subbasin only. Otherwise WaSiM will stop with an error message during initialization.
- After checking the DEM and the correct location of the lake, the water level-volume-relationship is established: Later, when balancing inflows, outflows, precipitation, evaporation a.s.o, the balanced volume must be used for re-calculating the water level. Since this is a non trivial problem, WaSiM has to create a look-up-table during initialization, otherwise, the inverse computation of the water level for a given volume would be extremely time consuming. To minimize computational costs (memory and CPU-time), this lookup-table works with three different default steps:
  - the uppermost 4 meters are subdivided into 1 cm-steps
  - the next 10 meters are divided into 2 cm-steps
  - the rest of the lake down to the ground is divided into 5 cm-steps.

Using this elevation-steps, the volume of the lake is now calculated for each elevation of the table. The result is a table which can be used for both, the computation of the volume for a given water level and the computation of a water level for a given volume. Both functions are implemented and used by the reservoir model. Also, two functions to calculate the reservoir volume from the POND-grid and to set the POND-values from the reservoirs balanced volume are vital to the algorithm.

An example of a waterlevel-volume-table is given here:

```
waterlevel-volume-table lake 1
Level[m a.s.l.]   Volume[m^3]
1917        3.85e+007
1916.99     3.84825e+007
1916.98     3.8465e+007
1916.97     3.84475e+007
1916.96     3.843e+007
1916.95     3.84125e+007
1916.94     3.8395e+007
1916.93     3.83775e+007
1916.92     3.836e+007
1916.91     3.83425e+007
1916.9      3.8325e+007
1916.89     3.83075e+007
```

… etc. down to 1913 m a.s.l. (4 m with 0.01 cm discretization)

```
1913.1      3.1675e+007
1913.09     3.16575e+007
1913.08     3.164e+007
1913.07     3.16225e+007
```

```
1913.06     3.1605e+007
1913.05     3.15875e+007
1913.04     3.157e+007
1913.03     3.15525e+007
1913.02     3.1535e+007
1913.01     3.15175e+007
1913        3.15e+007
1912.98     3.1465e+007
1912.96     3.143e+007
1912.94     3.1395e+007
1912.92     3.136e+007
1912.9      3.1325e+007
1912.88     3.129e+007
1912.86     3.1255e+007
1912.84     3.122e+007
1912.82     3.1185e+007
```

… etc. down to 1903 m a.s.l. (10 m with 0.02 cm discretization)

```
1903.2      1.435e+007
1903.18     1.4315e+007
1903.16     1.428e+007
1903.14     1.4245e+007
1903.12     1.421e+007
1903.1      1.4175e+007
1903.08     1.414e+007
1903.06     1.4105e+007
1903.04     1.407e+007
1903.02     1.4035e+007
1903        1.4e+007
1902.95     1.39125e+007
1902.9      1.3825e+007
1902.85     1.37375e+007
```

… etc. down to the lake bottom (with 0.05 cm discretization)

```
1895.6      1.05e+006
1895.55     962500
1895.5      875000
1895.45     787500
1895.4      700000
1895.35     612500
1895.3      525000
1895.25     437500
1895.2      350000
1895.15     262500
1895.1      175000
1895.05     87500
1895        0
```

For each lake, a file containing the watervelel-volume-table is written at model initialization to the working directory. The file name will be

W_V_Table_<nn>.dat → example: **W_V_Table_01.dat**

with nn = number of the lake (as coded in the lake-grid).

For initializing the water level and volume, there are two ways:

1. The initialization by read in grids. If the global readgrid-parameter or the readgrid-parameter for the lake model are set to 1, then the POND-grid is assumed to contain the values of a preceding model run. Then, the initial value of the lakes volume will be computed by using the waterlevel-volume-table with the water level taken from any POND cell of the lake (in fact, the first found cell is taken for this computation).

2. If readgrids is not active, then the initialization of the POND-grid is done by looking into the waterlevel-volume-table with a given volume (from routing description for SP-elements).

Similar approaches are used to initialize the tracer concentrations.

The last initialization step is the creation of the statistical balance file. The format of this file is described later. WaSiM will then do the initial routing (without changing the water level).

Now, the lake model is initialized completely.

During the simulation, for each time step the following scheme is processed (including the other sub models, for better consistency) :

- Potential evaporation is calculated depending on the land use parameters. If the cell is a lake cell and the POND grid contains water, then the default parameters for water surfaces will be used. Thus, it is not required any longer to have a land use or even a soil type for water surfaces.
  The potential and real interception evaporation and the potential and real evaporation from soil are set to zero.

- Within snow model, no snow accumulation will be permitted for lake cells with any water in the POND grid.

- In the unsaturated zone model (only Richards-version of WaSiM!), the POND-grid is processed as usual (with the restrictions described above, e.g. no interflow, no drainage, no exfiltration, no snow accumulation etc.).
  Groundwater exfiltration and re-infiltration from rivers as well as macro pore infiltration is suppressed for such cells, too.

- Finally, the routing model is executed. These are the different steps in this model, which are processed for all cells of a lake with water in the POND-cells (if a POND cell is dried, the cell belongs no longer to the lake and is not regarded here):
  - groundwater flow (infiltration into or exfiltration out from the uppermost soil layer) is estimated as mean value for all lake cells
  - precipitation is averaged for the entire lake
  - Evaporation losses are also averaged for the entire lake
  - However, also pond cells which are above the actual water level, may contain some water due to exfiltration (groundwater saturation) or precipitation. Such water is collected in the lake.
  - The volume of water from all active POND-cells together with the water volume from the saturated POND cells which are above the actual water level is then evenly redistributed across all lake cells in such a way, that the water level is identical for all cells. A slightly changed water level will result from this action. Cells above the new water level will now by dry - even if they had some exfiltrated groundwater or precipitation in its POND. Evaporation losses of those cells is still used for balancing the lakes water volume.
  - Now, that the exact starting volume for the time step is known, the actual area of the lake is calculated (surface area of all active POND cells is summed up)
  - Now, the balancing of water masses can start. At first, the irrigation amount and the abstractions are subtracted from the water volume.
    Note: if there are more than one reservoir or lake defined for a routing description by using SP-elements, only the last SP-element will be used for irrigation and abstractions.
  - The change in volume can now be computed: it is expressed by the following equation:

$$\Delta V = Q_{GW} - EP - Q_{irrig} - Q_{Abstr} + P \quad \textbf{(2.18.1)}$$

with
- ΔV change in volume [m^3]: positive values are net gains, negative values indicate a net loss of water for this time step
- Q_{GW} balance of groundwater flow (negative, if net loss, positive if net gain)
- EP evaporation from the lakes surface [m^3]
- Q_{irrig} irrigation water extraction [m^3]
- Q_{abstr} abstractions of water into other subbasins or external abstractions [m^3]
- P precipitation [m^3]

- o At this point, an error estimation is possible: since the old water value from the last time step is still available in the routing module, a comparison can be made between the difference of the new and the old water volume on the one side and the change in volume (as declared above) on the other side. Ideally, this would be zero. Usually, due to small rounding errors or numerical precision losses (e.g. when using the look-up-table with linear interpolation between the sampling points) the error value will be not zero. However, if an entity would be calculated incorrect, this error term would grow and indicate a balance error.

$$E = V_{act} - V_{old} - \Delta V \quad \textbf{(2.18.2)}$$

with
- E error estimation [m^3]
- V_{act} actual value as calculated in the unsaturated zone model
- V_{old} old water volume as known in the routing model in the last time step
- ΔV change in water volume as given in the formula above

- o Now, the tracer mixing of inflow and content as well as the change in concentrations due to evaporation and radioactive decay is computed.
- o Now, the abstraction rule (for regular outflow) is applied. The control file contains an [abstraction_rule_reservoir_<nn>]-section for each lake or reservoir. The format of this newly designed section is described later in this document. When applying the abstraction rule, also the inflow from upstream areas and/or other reservoirs and lakes is taken into account.
- o After subtracting the outflow from the volume (enlarged by inflow now but already reduced by abstractions, evaporation and irrigation), the new water volume is now distributed among the POND-cells of the lake, again resulting in a change in water level.
- o All parts of the water balance are now written to the statistics file. The entities are given in mm (with regard to the lake surface only). In m^3 and in m^3/s, where applicable. For details: see below.

Description of the new output file format for lake balances

For each lake, a file containing the water balance is written as statistics file. The filenames must be defined in the control file in the routing-model section in each SP-element-definition.
Example:

```
TG 1 (AE=1697.000, AErel=1.0)
 from OL 13 (kh=0.1, kv=0.4, Bh=41.9, Bv=167.4, Th= 4.19, Mh=25.0, Mv=10.0,
I=0.0010, L=18535.5, AE=1592.000)
  and SP 1  ( file=$outpath//Lake__01.//$year , V0 = 3.05E08, C0 = 1.6  0.1  3.0
0.0  0.0  0.0  0.0  0.0  0.0 )
```

```
and  AL 4 (modus = extern_with_rule $outpath//ableitung4.dat)
```

The filename of this example will resolve to e.g. d:\data\output\Lake_01.s84

The statistics file will contain the following columns:

Name of the column	description
YY	year as given is the meteorologic input data
MM	month as given is the meteorologic input data
DD	day as given is the meteorologic input data
HH	hour as given is the meteorologic input data
W[m]	water level in meter above sea level [m]
A[m2]	area of the lake (surface of POND cells which contain any water) [m^2]
V[m3]	water volume of the lake [m^3]
Qin[m3]	inflow of upstream subcatchments or other lakes [m^3]
Qin[mm]	inflow of upstream subcatchments or other lakes [mm]
Qin[m3s]	inflow of upstream subcatchments or other lakes [m^3/s]
Qout[m3]	outflow of the lake according to the abstraction rule [m^3]
Qout[mm]	outflow of the lake according to the abstraction rule [mm]
Qout[m3s]	outflow of the lake according to the abstraction rule [m^3/s]
Irrig[m3]	amount of irrigation water taken from the lake [m^3]
Irrig[mm]	amount of irrigation water taken from the lake [mm]
Irrig[m3s]	amount of irrigation water taken from the lake [m^3/s]
QGW[m3]	balance of groundwater inflow or outflow [m^3]
QGW[mm]	balance of groundwater inflow or outflow [mm]
QGW[m3s]	balance of groundwater inflow or outflow [m^3/s]
ETR[m3]	evapotranspiration losses [m^3]
ETR[mm]	evapotranspiration losses [mm]
ETR[m3s]	evapotranspiration losses [m^3/s]
PREC[m3]	precipitation yield [m^3]
PREC[mm]	precipitation yield [mm]
PREC[m3s]	precipitation yield [m^3/s]
dV[m3]	total change in volume during the last time step [m^3]
dV[mm]	total change in volume during the last time step [mm]
dV[m3s]	total change in volume during the last time step [m^3/s]
Abstr[m3]	abstractions from the lake (internal and external) [m^3]
Abstr[mm]	abstractions from the lake (internal and external) [mm]
Abstr[m3s]	abstractions from the lake (internal and external) [m^3/s]
Err[m3]	error estimation due to numerical inaccuracies and rounding [m^3]
Err[mm]	error estimation due to numerical inaccuracies and rounding [mm]
Err[m3s]	error estimation due to numerical inaccuracies and rounding [m^3/s]
<tracer symbol 1>	tracer concentration tracer 1
<tracer symbol 2>	tracer concentration tracer 2
..	
<tracer symbol n>	tracer concentration tracer n

The statistics file will be internally buffered like any other output file. To get intermediate results, Ctrl+C may be pressed and then code 2 may be entered to force a flush of the buffers to the hard disk.

Control file changes for lake modelling

There is a new section in the control file, called [lake_model].

Important parameters are the initialization method for the DEM and the default parameters for evaporation from open water surfaces.

```
[lake_model]
1                     # 0=ignore this module, 1 = run the module
2                     # method for recalculating DHM,
# 1 = do not change the DHM, it already reflects the ground of the lakes,
# 2 = use max_pond_grid to calculate dhm corrections; max_pond_grid will be
# used for mapping the cells pond content to a lake during model runs
0.15 # Albedo_OpenWater (will be used only, when the pond is filled with
# water when calculating potential evaporation -> otherwise,
# the normal landuse for this cell is referenced for this parameter)
0.1  #  z0_OpenWater (usage as above)
$readgrids            # readgrid code 0 do not read, 1 = read grids -->
# if 0, the initial value for the POND-grid as Volume of Lakes and
# Reservoirs is set by V0 from the routing description,
# if readgrids=1, no initialization in done (POND-Grid is read in)
# but the Vakt-Value is set by the various grids
```

Furthermore, the lake grid must be read in as a standard grid with the fillcode set to 0. The internal name of the lake grid is “lake_codes”, which must be given as identification string:

```
[standard_grids]
37                      # number of standard grids
...
$inpath//$lake_grid   lake_codes  0 # grid with a unique code for each lake
```

Input Grids required for lakes

The only required input grid is the lake grid. As described above, a lake consists of a group of 1..n cells sharing the same integer code greater than zero.

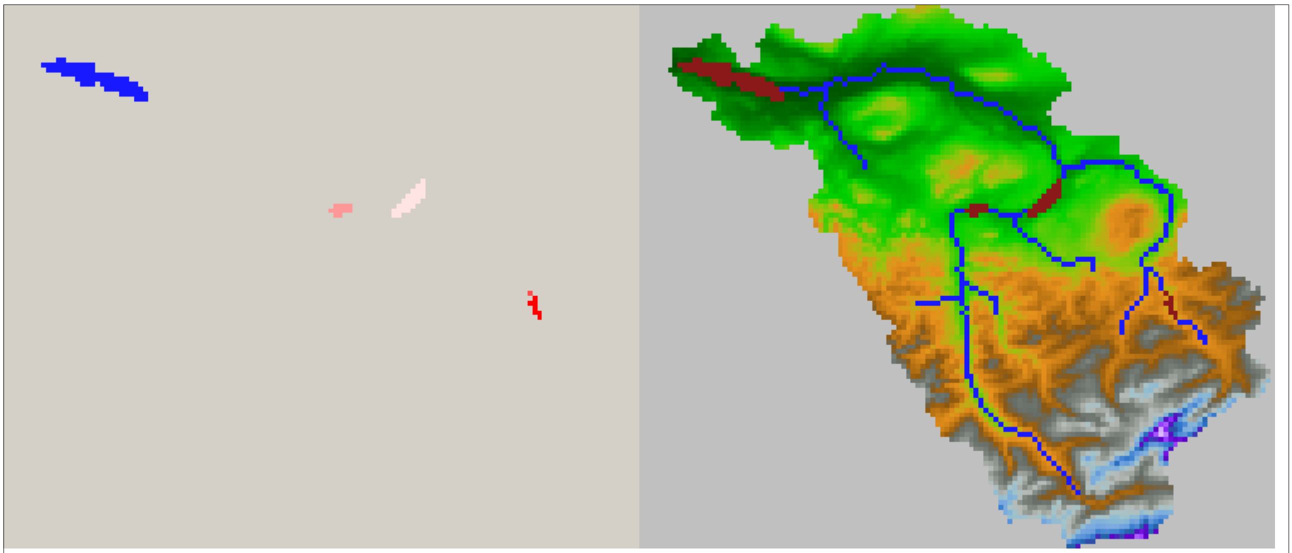

Figure 1: codes for lakes, here as a fictive example for the river Thur basin in Switzerland (these lake does not exist!). The right picture is only for better imagination, the left one is the real input grid.

The MAXPOND-grid (may also be known as POND-grid) must contain the maximum water depth-values for all lake cells in Meter. It is important, that the value of the DEM and the correspondent value of the MAXPOND grid all have the same sum (+/- 0.5cm) for the same lake.

At the same time the MAXPOND grid may contain the MAXPOND values for all other cells, too – but those cells will not be taken into account by the lake model, only by the unsaturated zones Richards model. So MAXPOND must contain the depth for the highest possible water level – including flooded areas around the lake. For an artificial reservoir, this is easy to achieve, but for natural lakes, this may be not so easy. To get the correct values for MAXPOND if only the mean water level is known, one should at first estimate the maximum (flooded) area and then look for the highest elevation within this area (which is assumed to be still dry at maximum water level). The difference between this maximum value and the actual elevation of all lake cells – which is always the lake ground! - describes the MAXPOND-value. Or, in other words: one must “create” a dam of the correct height (maximum water level) where the lake would pour normally and let the digital elevation model run through Tanalys. The elevation of the filled sink is then the maximum water of the lake. If DEM-data of the lakes ground are available, then MAXPOND is the difference between the filled sinks elevations and the lakes ground elevations. If only DEM-data of the mean lake surface are available, the depth data must be estimated elsewhere. After this operation, the surrounding cells which are not lake-cells should be checked for negative values (must be set to 0 or a positive number, if ponding should be regarded).

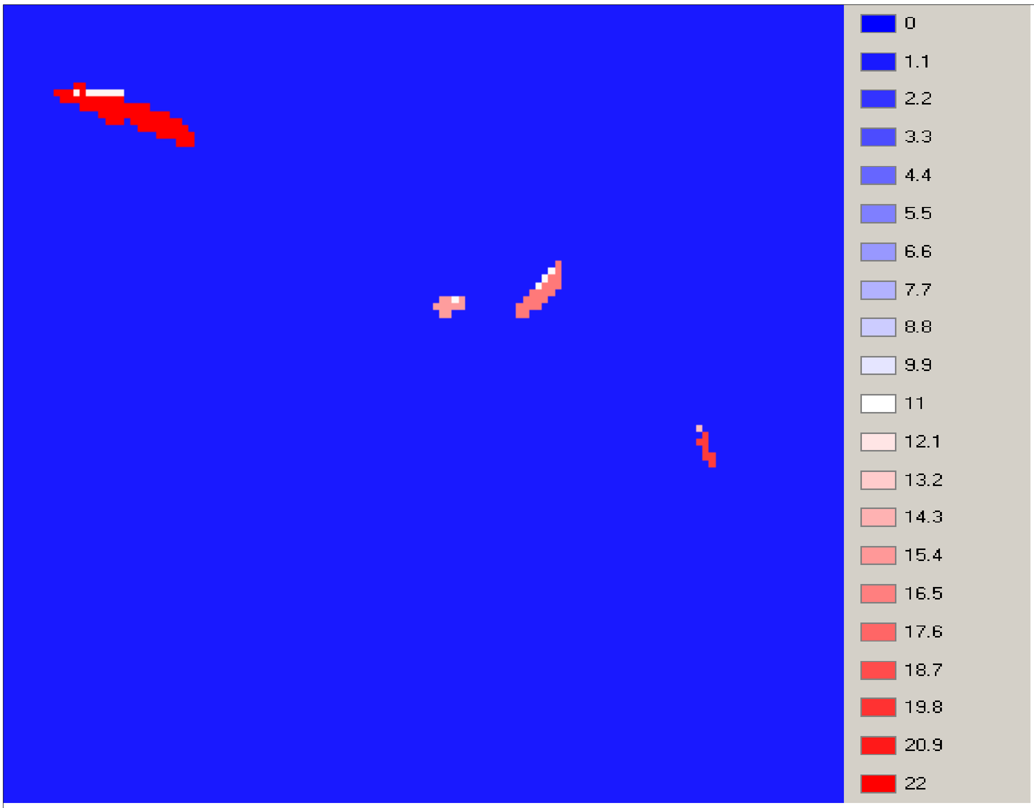

Figure 2: MAXPOND grid for the above lake grids from figure Fehler: Referenz nicht gefunden.

The MAXPOND grid must not contain any nodata values (at least not within the valid zones of the zone grid). All cells without ponds must be set to 0. Fehler: Referenz nicht gefunden shows an example for a MAXPOND grid. Shallow regions are visible by other colors, for example on the north shore of the greatest lake. The depth of 22m for most of the other cells is reduced to 11m for those cells on the north shore.

This example is an artificial example, no natural lake will have such uniformly distributed values. However, for modelling the lake within WaSiM, this kind of parameterization will be sufficient, since the horizontal resolution in this example is 500m, so the gradual descent of the lakes ground can not be modelled much better. But if the exact level of the ground of the lake is known, then this data can be used in the DEM directly. The method for the initialization of the lake model should then set to 1 (no change in DEM). Note, that the MAXPOND grid itself will be unaffected by the change of the initialization method. The value of DEM+MAXPOND, however, must again be identical for all cells of a lake.

Time variant abstraction rules

There are the following new features for abstractions:

- Abstractions can be read in from a file. The abstraction is taken from the river or the reservoir as given in the input file, the data have to be provided in m^3/s. The syntax of the file follows the rules for WaSiM table data input files (like for external inflows) but without tracer concentrations (and only one data column per file).
- An internal abstraction may take into account the maximum allowed discharge in the target channel (also called the targets capacity). When the capacity of the target (in m^3/s) is

reached, the abstraction will be leveled. The target capacity is an optional parameter.

- internal and external abstractions can now follow a time variable abstraction rule.

Control file format for abstraction rules

Abstractions may be defined in the following ways:

AL <code> (modus = intern <threshold> <fraction> <capacity> <measure>)

with	<code>	an unique integer identifier (ranging from 1 to number of abstractions)
	<threshold>	the minimum discharge in the routing channel for abstractions to start
	<fraction>	the fraction of the discharge exceeding the threshold to abstract [-]
	<capacity>	the maximum abstraction capacity
	<measure>	measure of threshold and capacity, usually as m^3/s

The internal abstraction needs a matching internal inflow, otherwise the model stops with an error message. The abstraction is taken from the main river of the actual routing description and put into the matching routing description of the target sub-basin (where a matching ZL description exists).

example: `AL 4 (modus = intern 1 1.0 2 m^3/s )`
`(somewhere in another routing description a definition of a ZL 4 (…) must exist)`

AL <code> (modus = extern <filename> <threshold> <fraction> <capacity> <measure>)

with <filename> the complete path and name of the result file to be written

All other parameters as described above. The external abstraction does not need a matching inflow, neither internally nor externally. The abstraction is taken from the main river of the target sub-basin.

Example: **`AL 1 (modus = extern $outpath//ableitung1.dat 0 0.01 1 m^3/s )`**

AL <code> (modus = extern_with_rule <filename>)

with filename defining the complete path (optional) and name of the result file

The difference between the "modus = extern" and "modus = extern_with_rule" is the usage of different rules: "extern_with_rule" does need a rules section for each abstraction (see below), whereas "extern" uses the parameters <threshold>, <fraction> and <capacity>. As for "modus = extern" and "modus = intern", the abstraction is taken from the main river of the target sub-basin.

Example: **`AL 4 (modus = extern_with_rule $outpath//ableitung4.dat)`**

(a section [abstraction_rule_abstraction_4] must exist in the control file to define the specific rules, see below)

AL <code> (modus = intern_with_rule)

Here, the abstraction is internal (will need an internal inflow with the same code), thus no output file is written. Whilst the abstraction with mode "intern" uses some parameters, this mode uses a specific rule definition as explained later.

Example: **`AL 3 (modus = intern_with_rule)`**

(a section [abstraction_rule_abstraction_3] must exist in the control file to define the specific rules, see below)

AL <code> (modus = from_file input_file = <inputfile> column=<c> output_file =

<outputfile>)

with	<code>	a unique integer identifier (ranging from 1 to number of abstractions)
	<inputfile>	the file the abstraction should be read from
	<c>	ID of the column containing the abstraction (starting from 5 → first 4 columns are for year, month, day and hour)
	<outputfile>	pateh and filename for the result file

Since the actual abstraction calculated in WaSiM may be limited compared to the abstraction in the inputfile (e.g. when the discharge in the routing channel is smaller than the read in abstraction), the outputfile contains the actual abstractions as computed in WaSiM.

Example: `AL 4 (modus = from_file input_file=$inpath//spende.84 column=5 output_file=$outpath//ableitung4.dat)`

AL <code> (modus = intern_with_rule_from_reservoir)

Here, the abstraction is taken from the reservoir which must be present in the same routing description instead of from the main river. A corresponding abstraction rule must exist. The difference between "modus = intern_with_rule" and "modus = intern_with_rule_from_reservoir" is the format of that specific rule: while the abstraction rule for "intern_with_rule" assumes discharges in m^3/s, the abstraction rule with type "intern_with_rule_from_reservoir" assumes reservoir content in m^3.

Example: `AL 78 (modus = intern_with_rule_from_reservoir )`

(a corresponding section [abstraction_rule_abstraction_78] must be defined in the control file, see below.)

For each abstraction which should have a time variant abstraction rule (all types with *with_rule* in it's type descriptor), a specific section in the control file is required, named according to the abstraction number [abstraction_rule_abstraction_<code>], with code matching the <code> from the AL-element definition.

Time variant abstraction rules are defined this way:

first row: number of following columns, followed by the Julian days for which rules will be established. For compatibility with old control files it is possible to have only one value in the first row indicating the number of following rows. Then, the number of rules is automatically set to one and only the first column of the rules table will be read in for the entire model time. The Julian days describe the LAST day the rule is valid for, so the year doesn't have to begin with 1 but may begin with 31 instead to indicate, that rule one is valid for the entire January. Also, the last JD doesn't have to be 366 - when no other rule follows the actual rule, the last rule is valid until the end of the year

following rows (according to the number given in the first row as first parameter): discharge in the routing channel [m^3/s], followed by the abstraction valid for this discharge [m^3/s] (one entry for each Julian day given in the first row)

some additional parameters can be defined for diurnal or weekly courses:
TargetCap: for each rule, the capacity of the abstraction can be limited (in case of input values overrun the maximum of the rule's allowed input and there is not as much capacity in the target of the abstraction (or in the pipeline the abstraction is carried away etc.).
WeekDays: for each rule the days of week (starting with 1 = Monday) can be defined, when the rule applies.

start_hour: at which time should the abstraction start (0=at midnight, 12=at noon, 21=in the evening at 21:00 etc.)
stop_hour: at which time should the abstraction end

Example:

```
[abstraction_rule_abstraction_4]
#           Julian Days; here: end of the months
12          32    60    91    121   152   182   213   244   274   305   335   366
7           0     0     0     0     0     0     0     0     0     0     0     0
8           1     0     0     0     0     0     0     0     0     0     0     1
9           2     1     0     0     0     0     0     0     0     0     1     2
10          3     2     1     0     0     0     0     0     0     1     2     3
11          4     3     2     1     0     0     0     0     1     2     3     4
14          7     6     5     4     3     3     3     3     4     5     6     7
15          7     7     6     5     4     4     4     4     5     6     7     7
16          7     7     7     6     5     5     5     5     6     7     7     7
17          7     7     7     7     6     6     6     6     7     7     7     7
18          7     7     7     7     7     7     7     7     7     7     7     7
27          7     7     7     7     7     7     7     7     7     7     7     7
27          8     7     8     7     8     7     8     7     8     7     8     7
TargetCap = 8     8     8     8     8     8     8     8     8     8     8     8
WeekDays  = 12345 12345 12345 12345 12345 12345 12345 12345 12345 12345 123 123
start_hour = 6    6     6     6     6     6     6     6     6     6     6     6
stop_hour = 18    18    18    18    18    18    18    18    18    18    18    18

[abstraction_rule_abstraction_78]
6           32    60    91    121   152   182   213   244   274   305   335   366
0           0     0     0     0     0     0     0     0     0     0     0     0
2.7625e08   0     0     0     0     0     0     0     0     0     0     0     0
2.925e08    2     2     2     2     2     2     2     2     2     2     2     2
3.0875e08   10    8     5     3     10    8     5     3     10    8     5     3
3.25e08     40    30    20    10    40    30    20    10    40    30    20    10
3.43e08     200   150   100   70    200   150   100   70    200   150   100   70
TargetCap = 8     8     8     8     8     8     8     8     8     8     8     8
WeekDays  = 67    67    67    67    67    67    67    67    67    67    67    67
start_hour = 6    6     6     6     6     6     6     6     6     6     6     6
stop_hour = 18    18    18    18    18    18    18    18    18    18    18    18
```

Notes:

- all abstraction rules must be enumerated from 1 to n without gaps.
- a single rule is one column of discharges for a given Julian day, mapped to the first column, the sampling points of the discharges in the routing channel
- between the sampling points of a rule a *linear interpolation* is carried out (but not between rules, i.e. there is no interpolation in time).
- If the discharge should jump suddenly, then a rule should contain the same sampling point twice:

  ```
  6
  7       0
  7       1
  8       1
  8       2
  9       2
  9       3
  ```

  The above example has to be read this way:
  There is no abstraction below 7 m^3/s in the routing channel. Then, between 7 and 8 m^3/s,

there will be a constant abstraction of 1 m^3/s, jumping to constant 2 m^3/s between 8 and 9 m^3/s and will stay at 3 m^3/s from 9 m^3/s discharge in the routing channel and more.

- There is *no interpolation* between the rules (e.g. for 14 m^3/s there will be an abstraction volume of 7 m^3/s until JD=32 and 6 m^3/s until JD=60 etc.)
- when plotting e.g. rule 1 for JD=32 in a diagram, one will see the monotonically increasing discharge between 7 and 14 m^3/s in the routing channel. This could also be defined like this:

```
[abstraction_rule_abstraction_4]
#           Julian Days; here: end of the months
12          32
7           0
14          7
27          7
27          8
TargetCap = 8
```

This would be perfectly fine for one rule. But since there are multiple rules, and all rules must share the same sampling points, it is required for all rules to have a value for each sampling point.

Time variant abstraction rules for reservoirs

The time variant abstraction rules for lakes and reservoirs (used for computing the outflow of the lake or reservoir) are very similar to the abstraction rules of abstractions from rivers:

```
[abstraction_rule_reservoir_1]
6           32    60    91    121   152   182   213   244   274   305   335   366
0           0     0     0     0     0     0     0     0     0     0     0     0
2.7625e08   0     0     0     0     0     0     0     0     0     0     0     0
2.925e08    2     2     2     2     2     2     2     2     2     2     2     2
3.0875e08   10    8     5     3     10    8     5     3     10    8     5     3
3.25e08     40    30    20    10    40    30    20    10    40    30    20    10
3.43e08     200   150   100   70    200   150   100   70    200   150   100   70
```

For all volumes above 3.43e08 m^3 the outflow equals the inflow in order to avoid an uncontrollable filling up of the lake. This example will be used in the following samples.

The main difference between time variant abstraction rules for real abstractions and for the outflow of a reservoir or lake are:

- the sampling points are given in m^3, not in m^3/s.
- There is no target capacity regarded
- no diurnal or weekly time dependence can be used here: WeekDays, start_hour and end_hour will be ignored.

The outflow measure is m^3/s.
The logic differences between an “abstraction_rule_abstraction_<n>” and an “abstraction_rule_reservoir_<n>” are:

- Each reservoir does need one and only one abstraction_rule_reservoir_n. This is outflow from the reservoir to the downstream area.
- Abstractions are optional and can be used for any target sub-basin. This is thought to be e.g. the water used for power generation, which is often channeled to other sub-basins

Samples of lake modelling with time variant abstraction rules

figure Fehler: Referenz nicht gefunden and Fehler: Referenz nicht gefunden show the Results for Lake 1 (the most downstream lake in the routing-scheme). For comparison, the results of a model run with the conventional simple lake model (lakes will be considered during routing only with no feedback to unsaturated zone, no evaporation and precipitation etc.) are also shown. As can be seen, the overall behavior of the graphs are very similar. Since the fully integrated lake model additionally takes the precipitation and the evaporation into account, the two graphs differ mainly during times with high precipitation (then, the new lake model may even result in a larger volume than the old one) and during times with high evaporation and without precipitation (then, the new lake model tends to slightly lower volumes). The effects are buffered because of the outflow rules, which are linearly interpolated between the sampling points: If the volume is already lower, than the outflow will be smaller, too, so in the next time step the difference in Volume will be smaller (if inputs are identical) than it would be if the outflow would be constant. This is a kind of negative feedback.

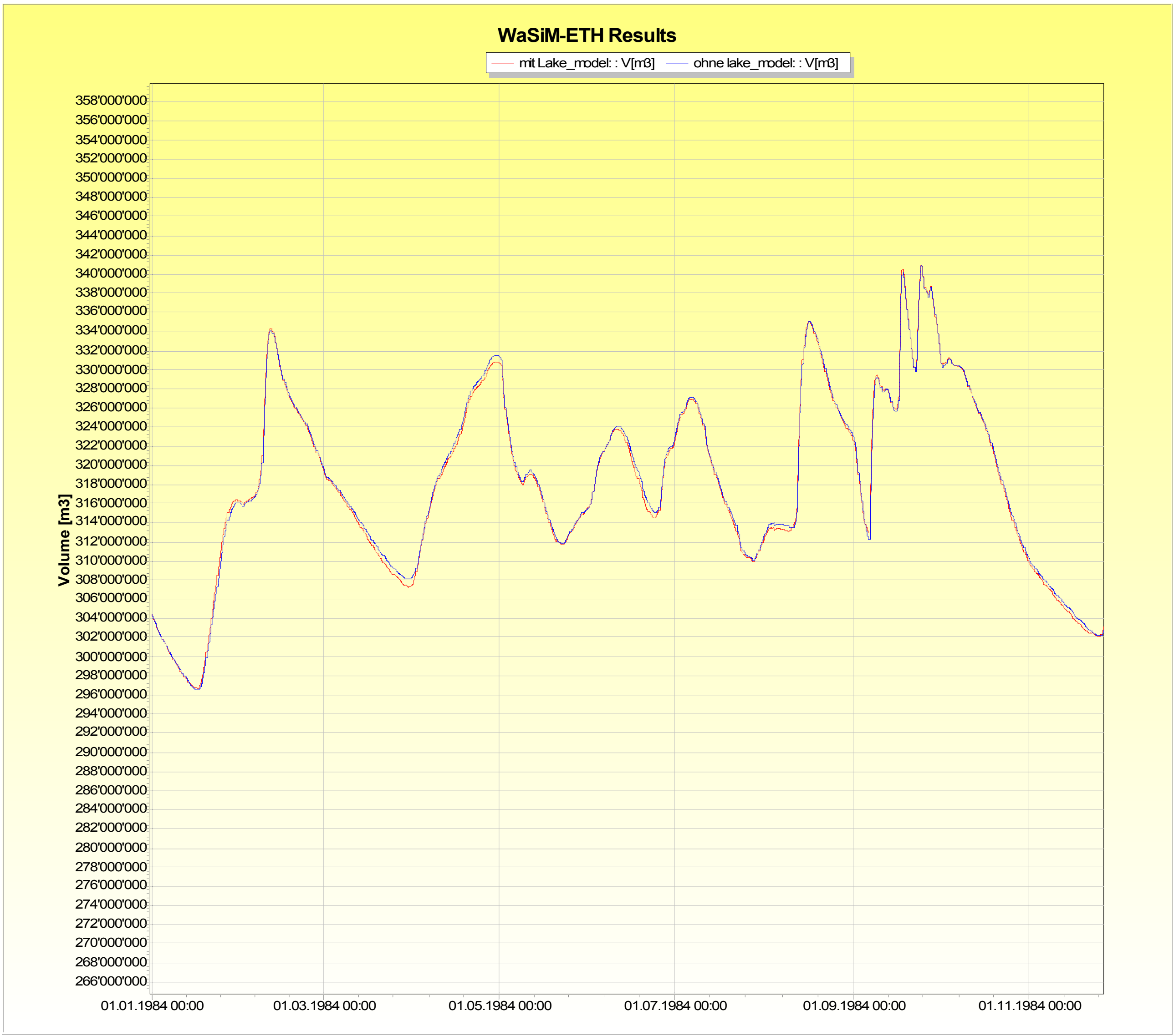

Figure 3: comparison of lake volume with the fully integrated lake model and with the simple (routing module only) lake model

In figure Fehler: Referenz nicht gefunden the other components of the water balance of lake 1 are shown.

- The inflow is drawn in red,
- the outflow is dark blue,
- the outflow from the comparison run with the old lake model is also dark blue but bold.
- Irrigation abstractions are dark green
- abstractions (which are defined using the AL-element) are dark grey
- precipitation is light blue
- evaporation is in light violet
- volume change (except inflow and outflow) is in light green
- error estimation is in dark violet

The interesting jumps in the outflow are the impact of the time variant outflow rule: On JD=182 and 213, the outflow for a given volume will be lowered considerably. At the same time, the abstraction rule for abstraction 4 (see description and example of abstraction rules) changes from 8 to 7 m^3/s (JD=182) and then back from 7 to 8 m^3/s (JD=213). It's obvious, that the average change in volume (expressed in m^3/s) equals the sum of the abstraction, the evaporation, the precipitation (counts negative) and the irrigation abstraction.

Groundwater-fluxes and error estimation are to small to be visible here.

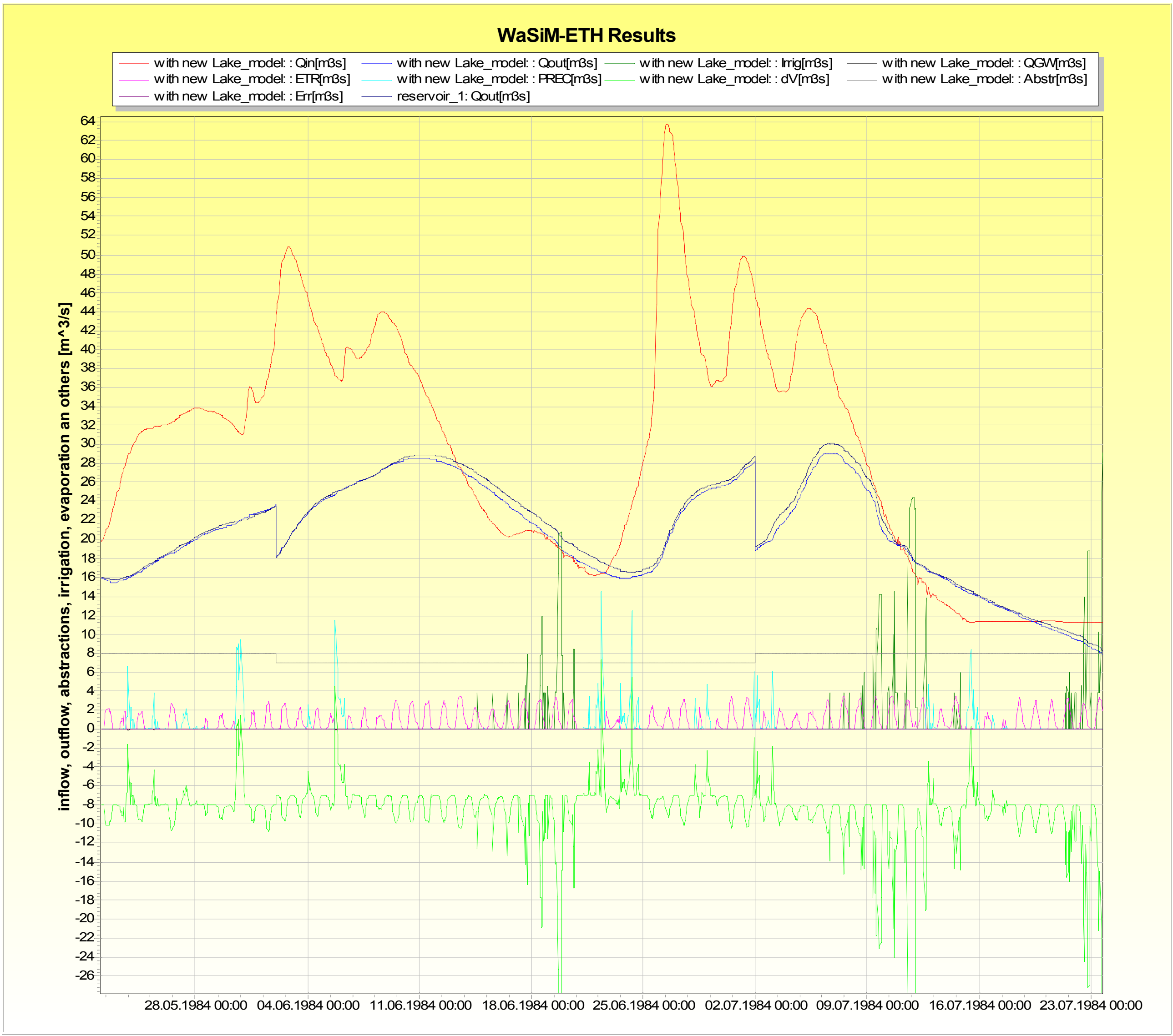

Figure 4: some components of the lakes water balance statistics.

During periods without rain but with higher evaporation values, the outflow with the new lake model is smaller than with the new one. The results could as well be shown in mm or in m^3.