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