Model Description

WaSiM

(Water balance Simulation Model)

last change: November 15, 2026 (version 11.02.04)

Hydrology Software Consulting J. Schulla
Regensdorferstrasse 162
CH 8049 Zürich
e-mail: j.schulla@wasim.ch

List of Content

1 Preface 15
2 Model description 16
2.1 Basic principles and main features 16
2.2 Correction of Precipitation 22
2.3 Interpolation of meteorological input data 26
2.3.1 Inverse Distance Interpolation (Method 1) 26
2.3.2 Elevation dependent regression (Method 2) 29
2.3.3 Linear combination of IDW + EDR (method 3) 34
2.3.4 Thiessen Polygons (method 4) 35
2.3.5 Bi-linear interpolation (method 5) 37
2.3.6 Linear combination of bi-linearly interpolated gradients and residuals (method 6) 38
2.3.7 Bi-cubic spline interpolation (method 7) 39
2.3.8 Linear combination of spline interpolated gradients and residuals (method 8) 40
2.3.9 Reading externally interpolated data as ASCII-grids, binary grids, GIF files or netCDF files (method 9) 41
2.3.10 Elevation dependent Regression with internal preprocessing (method 10) 47
2.3.11 linear combination of IDW and EDRINT (method 11) 47
2.3.12 Nearest neighbor combined with a constant lapse rate (method 12) 48
2.3.13 Using multiple interpolation methods at the same time (Regional Superposition) 49
2.3.14 Tolerating additional columns in meteorological input files 57
2.3.15 Optimized Smooth Interpolations for Larger Areas With Multiple Regions 58
2.4 Applying scenarios to the meteorologic driving data 61
2.4.1 Applying scenarios with low spatial resolutions 61
2.4.2 Applying monthly scenario grids 62
2.5 Topography dependent adjustment of radiation and temperature 62
2.5.1 Calculation of sun coordinates, sunset and sunrise, radiation correction 62
2.5.2 Modification of temperature 64
2.5.3 Control File Settings 65
2.5.4 Computing Relative Sunshine Duration from absolute Values 65
2.6 Evapotranspiration 65
2.6.1 Potential Evapotranspiration after Penman-Monteith 65
2.6.2 Evaporation from open Water 71
2.6.3 Energy Balance Aspects for interception evaporation, transpiration and soil evaporation 72
2.6.4 Potential Evapotranspiration after Wendling 73
2.6.5 Potential Evapotranspiration after Hamon 73
2.6.6 Potential Evapotranspiration after Haude 73
2.7 Control file settings for potential evapotranspiration 74
2.8 Calculating real evapotranspiration 75
2.8.1 WaSiM-version using the Topmodel-approach 75
2.8.2 WaSiM-version using the Richards-equation 75
2.8.3 Parameterization of single and multi-layer vegetation 76
2.8.3.1 [multilayer_landuse] table 76
2.8.3.2 [landuse_table] 77
2.8.4 Reducing real transpiration for wet conditions 78
2.9 Snow accumulation and melt 78
2.9.1 Snow accumulation 79
2.9.2 Classic snow melt approaches 79
2.9.3 Lateral, wind driven snow redistribution 81
2.9.4 Gravitational snow redistribution 84

2.9.5 Energy balance approach for snow melt....86
2.9.6 Control file settings for snow model....88
2.9.7 Output of Pure Snow Melt....89
2.9.8 Time variable snow melt coefficients for all T-index and T-u-index approaches....90
2.9.8.1 Introduction....90
2.9.8.2 Requirements in detail....90
2.9.8.3 Sample results....91
2.10 Interception....93
2.10.1 Classic bucket approach for rain....93
2.10.2 Control file settings for interception....95
2.10.3 Advanced interception of snow, including a canopy energy balance model....95
2.10.3.1 Snow canopy interception....95
2.10.3.2 Canopy energy balance....99
2.10.4 Control file extensions for advanced interception....104
2.11 Infiltration....106
2.11.1 Control file settings for the infiltration model....106
2.12 Soil model for the WaSiM version using the Topmodel-approach....106
2.12.1 Introduction....106
2.12.2 The course of water within the soil model....107
2.12.3 Control file setting for the soil model (Topmodel version)....110
2.12.4 Soil stability index (WaSiM-Topmodel-specific)....111
2.13 Silting up model and Surface routing-module....112
2.13.1 Key features of both modules....112
2.13.1.1 Silting-up Module....112
2.13.1.2 surface routing....112
2.13.2 Functional design....113
2.13.3 Theory of the Silting-up module....117
2.13.4 Control file extensions for Silting-up module....119
2.13.5 Theory of the Surface routing module....121
2.13.6 Control file extensions for Surface routing....131
2.13.7 Extension to surface routing module with 2D equation system....133
2.14 Soil model for the WaSiM version using the Richards-equation....136
2.14.1 Introduction....136
2.14.2 Parameterization....136
2.14.3 Numerical solution (vertical fluxes, interflow, infiltration/exfiltration in rivers)....137
2.14.4 Withdrawal of evaporating water....145
2.14.5 Considering irrigation....146
2.14.6 Considering artificial drainage....147
2.14.7 Considering clay horizons....148
2.14.8 Macropore infiltration....148
2.14.9 Considering ponding water....148
2.14.10 Groundwater table elevation....149
2.14.11 Groundwater recharge....150
2.14.12 Calculating baseflow....151
2.14.13 Coupling with the groundwater model....152
2.14.14 Dynamic time step control....152
2.14.15 Control file extensions for the unsaturated zone model....154
2.14.16 Optional Grids and Statistics for Field Capacity....156
2.14.17 A global parameter for defining the usable field capacity....161
2.14.18 Optional Output of Soil Moisture Within Root Zone as Relative Value....164
2.15 Modeling the groundwater flow and (lateral) transport....167
2.15.1 Flow equation....167
2.15.2 balance check when using boundary conditions....172

2.16 Discharge routing....174
2.16.1.1 Describing the routing structure in the control file:....177
2.16.2 Runoff Routing with Observed Values in m^3/s....181
2.16.3 Running the routing module in a shorter time step than the other modules....182
2.17 Routing only mode for WaSiM....186
2.17.1 Introduction....186
2.17.2 Requirements in detail....186
2.17.3 Sample results....187
2.18 Irrigation model....191
2.18.1 Optional Output of ETR2ETP Grid When no Irrigation is Active....192
2.18.2 control file options for using the different irrigation methods....193
2.19 Lake model....193
2.19.1 General mechanisms of the lake model....194
2.19.2 Impact of the lake model on the other sub models....194
2.19.3 How the lake model works....196
2.19.4 Description of the new output file format for lake balances....200
2.19.5 Control file changes for lake modelling....201
2.19.6 Input Grids required for lakes....202
2.19.7 Time variant abstraction rules....204
2.19.8 Time variant abstraction rules for reservoirs....208
2.19.9 Abstraction Rules for hydraulic connections between lakes....209
2.19.10 Samples of lake modeling with time variant abstraction rules....211
2.19.11 A new abstraction rule for lakes and reservoirs using externally provided runoff data 214
2.20 Performance criteria R^2 and EV....218
2.21 Coupling the solute transport to the water fluxes....219
2.22 Glacier model....220
2.22.1 Static glaciers....221
2.22.2 dynamic glaciers....224
2.22.3 Per-Sub-Basin-Parameterization for most glacier model parameters....229
2.23 Dynamic phenology....232
2.23.1 Dynamic Phenology 1 (DP1)....233
2.23.2 Dynamic Phenology 2 (DP2)....234
2.23.3 Dynamic Phenology 3 (DP3)....236
2.23.4 Dynamic Phenology 4 (DP4)....237
2.23.5 Considering the influence of soil moisture on phenology....238
2.23.6 Handling of phenological output data....239
2.24 Simple permafrost model....240
2.25 Temperature and heat transport model....241
2.25.1 Theory....241
2.25.2 Estimation of the relative saturation, S_E....242
2.25.3 Estimation of the effective thermal conductivity, $\lambda(T)$....243
2.25.4 Estimation of the effective hydraulic conductivity....245
2.25.5 Estimation of the effective heat capacity, $C(T)$....247
2.25.6 Estimation of the temperature shift for unsaturated soils....250
2.25.7 Modeling the temperature change....251
2.25.7.1 Boundary conditions....252
2.25.7.2 How does the numerical scheme work?....253
2.25.7.3 Solution schemes....255
2.25.8 Some Examples....258
2.25.8.1 Basics for applying the heat transfer model....258
2.25.8.2 Heat transfer example for different lower boundary conditions....264
2.25.8.3 Heat transfer example for unsaturated soils....267

2.25.8.4 An experiment with a very dry peat vs. a saturated peat....269
2.25.9 Setting up the heat transfer model in the control file....272
2.25.10 Additional remarks....274
2.26 Spin-Up for heat transfer model using HRU concepts....274
2.26.1 Motivation for a spin-up mode....274
2.26.2 The concept of HRUs....275
2.26.3 Setting up Spin-Up mode and HRU creation....276
2.26.4 Examples for spin-up results including soil temperature pre-spin-up....280
2.27 Layered Snow model....282
2.27.1 General considerations....282
2.27.2 Technical aspects....283
2.27.3 Sample results....284
2.27.4 Control file changes for layered snow model....287
2.28 Data Assimilation....289
2.28.1 Problem description....289
2.28.2 Grid-type data assimilation....290
2.28.3 Reservoir-type data assimilation....290
2.28.4 Control file extensions for data assimilation....291
2.29 Modeling Ice Wedge degradation and Changes in the arctic Vegetation with ArcVeg....293
2.29.1 Basic concepts of ArcVeg....293
2.29.2 Coupling ArcVeg with WaSiM – practical Guide....296
2.30 Automatic Soil Type Creation....309
3 Overview of required input data, parameters and data flow charts....315
3.1 Input data requirements for some model setups....315
3.2 Overview of the most important model parameters....318
4 Using WaSiM....322
4.1 Technical requirements - prerequisites....322
4.2 WaSiM runs from the command line with a control file....322
4.3 General structure of the control file....324
4.3.1 Structure of the control file....324
4.3.2 Using Comments....325
4.3.3 Using variables with the control file....325
4.3.4 Mandatory sections....326
4.4 Using the XML log file....327
4.5 Recommended Directory Structure....328
4.5.1 Input directory....328
4.5.2 Default Output directory....328
4.5.3 Initial State Directory....329
4.5.3.1 Seamless coupling of model runs with different time steps....330
4.5.4 Exchange Directory....332
4.6 Input file formats....332
4.6.1 Spatially gridded data....332
4.6.2 Meteorologic time series....334
4.6.3 Hydrological time series....336
4.7 Reading old states as initial conditions....336
4.7.1 Grids to be read....336
4.7.2 storage content file....338
4.8 Static and dynamic input grids....339
4.8.1 Standard grids....339
4.8.1.1 What grids can be standard grids....339
4.8.1.2 Options for standard grids (periodically reading, writing, statistics etc.)....342
4.8.2 variable grids....343
4.9 Defining outputs....345

4.9.1 statistics output....345
4.9.2 Grid output....347
4.10 Parametrizations of land use and soil types....349
4.10.1 Land use parameter (multilayer_landuse table and landuse_table)....350
4.10.1.1 Multi layer land use table....350
4.10.1.2 Basic land use table....351
4.10.2 soil parameters (multi-horizon soil table)....356
4.11 Using dynamic time steps in several modules....359
4.11.1 radiation correction....359
4.11.2 evaporation/transpiration....359
4.11.3 soil model....359
4.11.4 surface routing....361
4.11.5 Groundwater....361
4.11.6 discharge routing....361
4.11.7 lake model....361
4.12 Coupling with external models....362
4.13 Using variable cell sizes....365
4.14 Using shorter timesteps than one hour (minute time steps)....366
4.14.1 Introduction....366
4.14.2 Requirements in detail....367
4.14.3 Sample results....369
4.15 Sensitivity Analysis....373
4.15.1 Introduction....373
4.15.2 Parameters of the evaporation model....373
4.15.3 Parameters of the interception model....374
4.15.4 Parameters of the snow model....375
4.15.5 Parameters of the soil model for model version 1 (Topmodel-approach)....378
4.15.6 Parameters of the soil model for model version 2 (Richards-equation)....384
4.15.7 Spatial resolution....384
4.15.8 Conclusion....385
4.16 Calibrating the model....386
4.16.1 Introduction....386
4.16.2 Calibrating the soil model for model version 1 (Topmodel-approach)....386
4.16.3 Calibrating the soil model for model version 2 (Richards-equation)....393
5 Software Tools for Pre- and Postprocessing....397
5.1 Working with spatially distributed data (grids)....397
5.1.1 2DSPLINE....397
5.1.2 ASCIGRID....397
5.1.3 ASCII2STACK....397
5.1.4 FIELDGEN....398
5.1.5 FOCLMEAN....398
5.1.6 GRIDADD....398
5.1.7 GRIDASCI....398
5.1.8 GRIDCOLM....399
5.1.9 GRIDDIFF....399
5.1.10 GRIDEDIT....399
5.1.11 GRIDGAUS....399
5.1.12 GRIDGRID....399
5.1.13 GRIDMASK....400
5.1.14 GRIDMULT....400
5.1.15 GRIDQUOT....400
5.1.16 GRIDRAND....400
5.1.17 GRIDSTAT....401

5.1.18 GRIDSURF....401
5.1.19 MAKEGRID....401
5.1.20 MF2WASIM....401
5.1.21 RECLASS....401
5.1.22 REFINE....401
5.1.23 REMAP....402
5.1.24 RESAMPLE....403
5.1.25 RESIZE....403
5.1.26 SURFGRID....403
5.1.27 TANALYS....403
5.1.28 TOPOFACT....409
5.1.29 ZONESTAT....410
5.2 Working with station data (meteorological time series)....410
5.2.1 IDWP....410
5.2.2 REGRESS....411
5.2.3 REGR....411
5.2.4 REGRASCI....412
5.2.5 ASCIREGR....412
5.2.6 GMDtoTAB....412
5.2.7 QtoSPEND....412
5.2.8 SPENDINV....412
5.2.9 SONNEREL....413
5.3 Software for visualization....413
5.3.1 REGRESS....413
5.3.2 SHOWGRID....413
5.3.3 GRAPHLINES....415
5.4 Other programs....416
5.4.1 MAKEGWN....416
5.4.2 RESAGGR....416
5.4.3 RESAGG....416
5.4.4 RESMEAN....416
5.4.5 RESMEANI....417
5.4.6 QUANTIL....417
5.4.7 LOWFLOW....417
5.4.8 FLOODS....417
6 References....418
7 Appendices....421
7.1 Example of a control file for WaSiM....421
7.2 Example of a control file for Tanalys....441
7.3 Error codes of WaSiM....444
7.4 Internal names for grids that can be used for writing special output....449

List of Tables

Table 2.1.1: WaSiM main features..... 17
Table 2.6.1: monthly differences between day and night average temperatures, referring to sea level, valid for northern Switzerland.....71
Table 2.6.1: Monthly correction factors f_i for Hamon-Evapotranspiration (valid for northern Switzerland).....73
Table 2.6.2: Monthly correction factors k_M for Haude-Evapotranspiration (valid for Germany).....74
Table 2.13.1 : coefficients for calculating roughness coefficient M for conserving cultivation.....127
Table 2.22.1: Comparison of phenological models implemented into WaSiM.....233
Table 3.1.1: required submodels and model-input data for different model objectives.....315
Table 4.8.1: list of standard_grids identification codes.....339
Table 4.8.2: list of variable_grids identification codes.....344
Table 4.10.1: Reference table of the parameters for an entry in the multilayer_landuse table.....350
table 4.10.2: Reference table for all possible parameters for a land use table entry.....351
table 4.10.3: Reference table for all possible parameters for a soil table entry.....356

List of Figures

figure 2.1.1: model structure of WaSiM....16
figure 2.2.1: Interpolation regions for Switzerland. The regions of interest for the examples are green and blue/green shaded with numbers 13, 16 and 130....23
figure 2.2.2: interpolated precipitation with uniform precipitation correction for all regions....24
figure 2.2.3: interpolated precipitation with uniform precipitation correction for all regions....24
figure 2.3.1.1: example for IDW interpolation of precipitation (January 3, 1994, 18:00)....28
figure 2.3.1.2: example for EDR interpolation for the same data as in figure 2.3.1.1....28
figure 2.3.2.1: typical temperature gradients (here for May 5, 1996, hours 1 to 12)....32
figure 2.3.2.2: Example for elevation dependent regression (January, 1, 1984, 14:00)....33
figure 2.3.3.1: example for a combination of IDW + EDR (January 3, 1984, 17:00), same data as for examples for method 1....35
figure 2.3.4.1: example for interpolation using Thiessen polygones (precipitation, January 3, 1994, 17:00)....36
figure 2.3.5.1: scheme of calculating weights for bilinear interpolation to point i,j....37
figure 2.3.9.1: Example for a Swiss radar data image covering entire Switzerland with three radar stations (unmapped values)....43
figure 2.3.9.2: interpolated region of interest, mapped to precipitation values....43
figure 2.3.9.3: Sample correction factor grid for radar data for the river Thur basin....44
figure 2.3.13.1: Weighting of multiple regions for a point p near the regions crossing point....51
figure 2.3.13.2Weighting of the actual region if using multiple regions. Rheinland-Pfalz, Germany, 5 regions, maximum transition distance r was set to 20 km....53
figure 2.3.13.3: Definition of two regions for the river Thur basin (Switzerland, 1700km2), the entire grid encompasses an area of 65.5 km x 56 km, each cell is 500 by 500m (also for all following figures)....56
figure 2.3.13.4: interpolation results for IDW (North) and EDR (South) without smooth transition 56
figure 2.3.13.5: interpolation results for IDW (North) and EDR (South) with 1km transition-range56
figure 2.3.13.6: interpolation results for IDW (North) and EDR (South) with 5km transition-range57
figure 2.3.13.7: interpolation results for IDW (North) and EDR (South) with 10km transition-range 57
figure 2.3.15.1: Example of relative air humidity interpolation results for all Swiss river basins, interpolated independently of each other....58
figure 2.3.15.2: Example of relative air humidity interpolation results for all Swiss river basins, interpolated independently of each other but with smooth shifts between regions....59
figure 2.8.4.1: Interdependence between transpiration and soil water content. PAW: plant-available water, DW: drainable or gravitational water, PWP: permanent wilting point, FC: field capacity, Sat: soil water content at saturation, HReduDry: threshold value for starting dryness stress, TReduWet: threshold value for starting oxygen stress due to (nearly) water saturated soils, LimitReduWet: maximum reduction of transpiration due to oxygen stress....78
figure 2.9.3.1: Directed sky view factor (SVFdir) for 180° to 270° (i.e. SW) for the central Swiss Alps....82
figure 2.9.3.2: wind driven snow correction factors c_wind for the central Swiss Alps region. Average: 0.028, min/max:-0.7, 1.04; profile A-B: see figure 2.9.3.3....83
figure 2.9.3.3 profile A-B (see figure 2.9.3.2) for a SW main wind direction....84
figure 2.9.4.1: snow distribution pattern without slides (in mm SWE)....85
figure 2.9.4.2: snow distribution pattern with slides (in mm SWE)....86
figure 2.9.8.1: comparison of modeled runoffs versus observed runoff (red) for blue=constant T-u-index parameters and green = temporally variable parameters....92
figure 2.9.8.2: Comparison of observed runoff (red) vs. modelled runoff (blue) for default static snow melt parameters....92
figure 2.9.8.3: Comparison of observed runoff (red) vs. modelled runoff (blue) for temporally

variable snow melt parameters..... 93
figure 2.10.3.1: Specific SWE stored in the catchment..... 104
figure 2.10.3.2: Comparison of simulated and measured discharge data during validation run in winter 2017/18..... 104
figure 2.13.2.1: Flow chart for traditional soil- and routing modelling (prior to version 8.1.01 and without silting-up and surface routing since version 8.1.01)..... 114
figure 2.13.2.2: Flow chart for soil- and routing modelling including silting-up and surface routing (starting from version 8.1.01)..... 115
figure 2.13.2.3: Flow chart for soil- and routing modelling including surface routing but without silting-up..... 116
figure 2.13.2.4: Flow chart for soil- and routing modelling including silting-up but without surface routing..... 117
figure 2.14.16.1: Residual water content in minFieldCapacity_mm..... 159
figure 2.14.16.2: Maximum usable field capacity in mm..... 160
figure 2.14.16.3: actual field capacity 1991-01-31 in mm..... 160
figure 2.14.16.4: actual water content 1991-01-31 as percentage of actual content on field capacity (in %)..... 161
figure 2.14.18.1: statistics for relative soil moisture for the river Thur basin, January 1991..... 165
figure 2.14.18.2: spatial distribution of relative soil moisture for January, 7th, 1991, Thur basin. .166
figure 2.14.18.3: spatial distribution of relative soil moisture for January, 31st, 1991, Thur basin. 166
figure 2.17.3.1: Comparison between routing results for a normal model run (red) and a routing only model run (blue) with identical parameters; 2018, Gauge Kander/Hondrich (491km2)..... 188
figure 2.17.3.2: Comparison between routing results for a normal model run (red) and a routing only model run (blue) with identical parameters; December 2018 flood event; Gauge Kander/Hondrich (491km2)..... 189
figure 2.17.3.3: Comparison between routing results for a normal model run (red) and a routing only model run (blue) with different roughness parameters; 2018, Gauge Kander/Hondrich (491km2)..... 189
figure 2.17.3.4: Comparison between routing results for a normal model run (red) and a routing only model run (blue) with different roughness parameters; December 2018 flood event; Gauge Kander/Hondrich (491km2)..... 189
figure 2.17.3.5: conmparison between original model run and routing only model run for the outflow of Lake Thun, 2018; red=normal model run, blue=routing only model run; green=precipitation from the lake surface, black = evaporation from the lake surface..... 190
figure 2.17.3.6: Comparison between original model run and routing only model run for the content of Lake Thun, 2018; red=normal model run, blue=routing only model run..... 191
figure 2.19.6.1: Codes for lakes, here as a fictive example for the river Thur basin in Switzerland (these lakes do not exist!). The right picture is only for better imagination, the left one is the real input grid..... 203
figure 2.19.6.2: MAXPOND grid for the above lake grids from figure 2.19.6.1..... 204
figure 2.19.9.1: example for hydraulic connections between three lakes..... 209
figure 2.19.9.2: example for flow reversals between lakes Bielersee and Neuenburgersee due to the new abstraction rule as hydraulic connection..... 210
figure 2.19.9.3: Example for modeled lake water table with and without the new hydraulic connection abstraction rule (red=observed water table, green=modeled water table without hydraulic connection, i.e. classic one-way-routing, and blue=modeled water table with hydraulic connection)..... 211
figure 2.19.10.1: comparison of lake volume with the fully integrated lake model and with the simple (routing module only) lake model..... 212
figure 2.19.10.2: some components of the lakes water balance statistics..... 213
figure 2.22.2.1: Glaciers in the Swiss Alps (center: Aletschgletscher) for 1996..... 229
figure 2.22.2.2: Glaciers in the Swiss Alps for 2010..... 229
figure 2.25.2.1: Change in the fraction of liquid water, S E, during temperature change in a sandy

loam soil (alpha=2, n=2, m=0.5).(temperatures are negative from -10 to 0°C).....243
figure 2.25.3.1: Effective thermal conductivity plotted against soil temperature for different relative water contents Θ rel (1, 0.8, 0.5 and 0.2).....244
figure 2.25.4.1: Effective thermal conductivity based upon Eq. (2.24.8a).....245
figure 2.25.4.2: Effective hydraulic conductivity using an exponential function (Eq. 2.24.8b).....246
figure 2.25.4.3: Effective unsaturated hydraulic conductivity using an exponential function and a variable ks-value based on equation (2.24.8c).....247
figure 2.25.5.1: The change in energy, i.e. latent heat, per temperature change of 0.01°C for a soil with Θ s = 0.5 and Θ r = 0.1 for saturated (Theta=0.5) and unsaturated (Theta=0.18) conditions..249
figure 2.25.5.2: The change in effective heat capacity of a freezing/thawing soil (Θ s = 0.5 and Θ r = 0.1) for saturated (Theta=0.5 and unsaturated (Theta=0.18) conditions.....250
figure 2.25.7.1: Definition of layer indices for the discrete 1D heat transfer model describing heat transfer through conduction.....252
figure 2.25.8.1: temperatures used for heat transfer test data for hourly data (daily fluctuation +/- 10°C).....260
figure 2.25.8.2: soil temperatures for the artificial test site in hourly reslution (shown are temperatures for 10 cm to 110 cm in 10 cm steps and temperatures in 2, 3, 4, 7 and 8.8m).....260
figure 2.25.8.3: temperatures used for heat transfer test data for daily data.....261
figure 2.25.8.4: soil temperatures for the artificial test site (upper horizon: 20 cm of organic material, modelled in 10 layers of 2 cm each, then 10 cm layers follow; profile has a total depth of 10 m --> the fat, dark grey graph is at around 7 m).....261
figure 2.25.8.5: Thaw depth for simulation in hourly and daily resolutions (red: hourly resolution with SEthreshold = 0.5; blue: daily resolution with SEthreshold = 0.5; grey: hourly resolution with SEthreshold = 0.8).....262
figure 2.25.8.6: Model results from an artificial arctic test site with a Barrow-like climate without the soil heat transfer model. The curves represent observed runoff (green), total modeled runoff (red), and modeled interflow (blue).....263
figure 2.25.8.7: Model results from an artificial arctic test site with a Barrow-like climate with the soil heat transfer model. The curves represent observed runoff (green), total modeled runoff (red) and modeled interflow(blue).....263
figure 2.25.8.8: air temperature for the following experiments. T ranges from -25°C in winter to +5°C in summer with diurnal fluctuations of +/- 10°C. Annual average is -10°C.....264
figure 2.25.8.9: soil temperatures as time series for constant T lower boundary condition (red=0.7m, blue = 1.7m, green = 2.7m, black = 3.7m etc. down to 8.8m for violet).....264
figure 2.25.8.10: soil temperatures as time series for constant heat flux of 65mW/m^2 lower boundary condition (red=0.7m, blue = 1.7m, green = 2.7m, black = 3.7m etc. down to 8.8m for violet).....265
figure 2.25.8.11: T-Profile with constant T -9.478°C as lower boundary condition.....266
figure 2.25.8.12: T-profile with initial T of -9.478°C and constant heat flux of 65mW/m^2 as lower boundary condition.....267
figure 2.25.8.13: Comparison of soil temperatures in 20 cm depth for a saturated (blue) and unsaturated (red) peat (porosity 0.8);.....268
figure 2.25.8.14: Comparison of soil temperatures in 45 cm depth for a saturated (blue) and unsaturated (red) soil with 20 cm peat (porosity 0.8) above 25 cm silty clay.....269
figure 2.25.8.15: Comparison of soil temperatures in 100 cm depth for a saturated (blue) and unsaturated (red) soil with 20 cm peat (porosity 0.8) above 80 cm silty clay.....269
figure 2.25.8.16: temperature time series in a saturated (blue) and unsaturated (red) soil in 1m depth; the soil has a 1m peat layer on a silt. For dry peat, the groundwater is at about -1.3m. No rain, no snow, no evaporation has been regarded in this experiment, only temperature as given in the examples above.....270
figure 2.25.8.17: T-profiles for saturated soil.....271
figure 2.25.8.18: T-profiles for unsaturated soil.....272
figure 2.26.3.1: HRUs for all cells (basin of the river Aare up Berne, Switzerland, 11826 cells

(=2956.5 km2)..... 278
figure 2.26.3.2: HRU master cells (basin of the river Aare up Berne, Switzerland, 2426 cells.....278
figure 2.26.3.3: All HRUs for a flat arctic tundra basin with typical permafrost structures (low centered polygons).....279
figure 2.26.3.4: HRU master cells for a flat arctic tundra basin.....279
figure 2.26.4.1: soil temperature profiles without pre-spin-up (solid lines) and with pre-spin-up (dashed lines..... 281
figure 2.27.3.1: soil temperatures for the first four soil layers (each 0.33m thick) with and without layered snow model.....284
figure 2.27.3.2: average thaw depth for the same high alpine sub basin using layered snoe model (blue) and using the conventional snow model (red).....285
figure 2.27.3.3: SWE solid and liquid per layer as special output time series.....286
figure 2.27.3.4: Detail of SWE solid and liquid per layer. The thin graphs are the liquid water content..... 286
figure 2.27.3.5: snow temperatures for a melting snow pack with up to eight snow layers.....287
figure 2.29.2.1 IceWedgeLowerLimit (iwdglo) for the test basin (depth in cross section is between 2 and 4 m below surface).....300
figure 2.29.2.2 IceWedgeUpperLimit (iwdgup) for the test basin (depth in cross section is at 0.1 m below surface for all calls on an ice wedge).....301
figure 2.29.2.3: Excess ice stack: Level 3, 9, 14 and 17 with 45.9, 41.8, 183.6 and 183.6 mm WE 301
figure 2.29.2.4: original DEM (almost) without any channels.....307
figure 2.29.2.5: DEM after year 1, the cross section shows the decreased soil surface over the degraded ice wedges.....307
figure 2.29.2.6: time series for geodetic elevation of the uppermost 8 soil layers during ice wedge degradation (1999) and the following stabilazation phase (2000 and 2001).....308
figure 2.30.1: sand fraction as a medium value for 0...120 depth, ranging from around 2% (brown) to 70% (blue)..... 309
figure 2.30.2: silt fraction as a medium value for 0...120 depth, ranging from around 14% (brown) to 70% (blue)..... 310
figure 2.30.3: clay fraction as a medium value for 0...120 depth, ranging from around 30% (brown) to 85% (blue)..... 311
figure 2.30.4: USDA Soil texture triangle (taken from https://drainagematters.org/stormwater-glossary/usda-soil-texture-triangle.html, 2026-11-05).....311
figure 4.5.3.1: Effect of new initialization of runoff concentration and runoff routing when coupling runs with different time steps (here: green uses 1h, black uses 10min time step).....331
figure 4.14.3.1: comparison of modelled runoff for a 29 km2 subbasin of the river Kander (Canton of Berne, Switzerland) for 2010, red is for time step 10 minutes, blue for time step 1 hour, precipitation network is identical.....369
figure 4.14.3.2: comparison of modelled runoff for a 29 km2 subbasin of the river Kander (Canton of Berne, Switzerland) for 2015, red is for time step 10 minutes, blue for time step 1 hour.....370
figure 4.14.3.3: comparison of modelled and observed runoff for a 29 km2 subbasin of the river Kander April/May 2015. Time series in green and black are observed runoff data with time step 10 minutes (green) and 1h (black), red (10min) and blue (1h) are modelled runoffs.....371
figure 4.14.3.4: comparison of modelled and observed runoff for a 29 km2 subbasin of the river Kander May 1st to May 2nd, 2015. Time series in green and black are observed runoff data with time step 10 minutes (green) and 1h (black), red (10min) and blue (1h) are modelled runoffs.....371
figure 4.14.3.5: comparison of modelled and observed runoff for a 29 km2 subbasin of the river Kander July 2015. Time series in green and black are observed runoff data with time step 10 min (green) and 1h (black), red (10min) and blue (1h) are the modelled runoffs.....372
figure 4.14.3.6: comparison of modelled and observed runoff for a 68 km2 subbasin of the river Aare, June 2015. Time series in green and black are observed runoff data with time step 10 min (green) and 1h (black), red (10min) and blue (1h) are the modelled runoffs.....372

figure 4.15.2.1: model sensitivity on changes in the surface resistances rs, using results from the Rietholzbach basin for 1984; units of ETP and discharge are in mm/day.....374
figure 4.15.4.1: Snow accumulation at various threshold temperatures TR/S (snow/rain) (a) Ttrans = 0.0 K; (b) Ttrans = 2.0 K, Rietholzbach catchment, 1984 “Schneespeicher” means snow water equivalent; “Niederschlag” means precipitation.....376
figure 4.15.4.2: temporal course of snow water equivalent for different degree day factors: (a) melt threshold temperature T0,m = +0.5 °C, (b) T0,m = -1.5 °C, Rietholzbach basin, 1984; “Schneespeicher” means snow water equivalent, “Niederschlag” means precipitation.....377
figure 4.15.4.3: temporal course of snow water equivalent for different melt threshold temperatures T 0,m at different degree day factors, Rietholzbach basin, 1984; “Schneespeicher” means snow water equivalent, “Niederschlag” means precipitation.....377
figure 4.15.5.1: model sensitivity on variations in Tkorr if using a too small recession parameter m (15 mm), showed for (a) soil moisture, (b) saturation deficit and (c) discharge; Wernersbach, Saxonia, 4.6 km2, 365 to 465 m a.s.l., hourly values, 1993; Abfluss=discharge, Sättigungsdefizit = saturation deficit, Bodenspeicher=soil moisture, beobachtet=observed.....379
figure 4.15.5.2: model sensitivity on variations of Tkorr at a optimum recession parameter m (55 mm), showed for (a) soil moisture, (b) saturation deficit and (c) discharge; Wernersbach, Saxonia, 4.6 km2, 365 to 465 m a.s.l., hourly values, 1993, Abfluss=discharge, Sättigungsdefizit = saturation deficit, Bodenspeicher=soil moisture, beobachtet=observed.....380
figure 4.15.5.3: distribution of model efficiencies (R2) as function of Tkorr and m; Wernersbach, Saxonia, 4.6 km2, 365 to 465 m a.s.l., daily time step, 1993.....381
figure 4.15.5.4: Impact of the threshold precipitation intensity Pgrenz for generating macro pore drainage on (a) soil moisture, (b) saturation deficit and (c) runoff; without capillary rise/reflow from interflow into soil storage (rk = 0.0); Wernersbach, Saxonia, 4.6 km2, 365 to 465 m ü.M., daily time steps for 1993; “Abfluss”=discharge, “Sättigungsdefizit”=saturation deficit, “Bodenspeicher”=soil storage, “beobachtet”=observed.....382
figure 4.15.5.5: Impact of the threshold precipitation intensity Pgrenz for generating macro pore drainage on (a) soil moisture, (b) saturation deficit and (c) runoff; without capillary rise/reflow from interflow into soil storage (rk = 0.0); Wernersbach, Saxonia, 4.6 km2, 365 to 465 m ü.M., daily time steps for 1993; “Abfluss”=discharge, “Sättigungsdefizit”=saturation deficit, “Bodenspeicher”=soil storage, “beobachtet”=observed.....383
figure 4.15.7.1: Impact of the spatial resolution on the model efficiency, resolution varied between 200 m and 10’000 m, 1 year simulation of discharge, time step 1 hour, the upper graph for the entire Thur basin, the lower graph for the more alpine upper Thur basin at Stein/Thur.....385
figure 4.16.2.1: Estimation of the recession parameter m of the river Verzasca (Switzerland) for 1993.....387
figure 4.16.2.2: 1st optimizing step: distribution of the model efficiency R2 (linear and logarithmic) as reaction on variations in Tkorr; Verzasca, Simulation in daily time steps for 1993.....390
figure 4.16.2.3: 2nd optimizing step: distribution of the model efficiency R2 (linear and logarithmic) as reaction on variations in Tkorr; optimum for Tkorr at around 0.25 for R2(log); Verzasca, Simulation in daily time steps for 1993.....390
figure 4.16.2.4: model sensitivity on variations in Tkorr; Verzasca, daily time step for 1993; “Abfluss”=discharge, “Sättigungsdefizit"=saturation deficit,”Niederschlag”=precipitation.....391
figure 4.16.2.5: distribution of model efficiencies (R2) following variations in m and Tkorr; optimum values at app. Tkorr = 0.14, m = 0.036 m; Verzasca, 1993.....392
figure 4.16.3.1: Modeled runoff using model version 2 for the Rietholzbach catchment, without interflow.....393
figure 4.16.3.2: Modeled runoff using model version 2 for the Rietholzbach catchment, with interflow, krec = 0.9.....394
figure 4.16.3.3: Modeled runoff using model version 2 for the Rietholzbach catchment, with interflow, krec = 0.3.....395
figure 4.16.3.4: Modeled runoff using model version 2 for the Rietholzbach catchment, with interflow, krec = 0.1.....395

figure 4.16.3.5: Modeled runoff using model version 2 for the Rietholzbach catchment, krec = 0.3, dr = 12.....396
figure 4.16.3.6: Modeled runoff using model version 2 for the Rietholzbach catchment, krec = 0.1, dr = 30.....396
figure 5.1.27.1: topographic analysis of a digital elevation model by TANALYS.....404
figure 5.1.27.2: ky-view-factor ψ sky and fraction of the sky As depending on the horizon overhead angle □ h.....406
figure 5.1.27.3: flow direction correction at mouths.....407
figure 5.1.27.4: flow travel time grid for the river gauge network of the Thur-basin, hour zones...409
figure 5.3.2.1: screenshot of ShowGrid.....414
figure 5.3.3.1: Example for Graphlines.....415

1 Preface

This completely revised model description of 2012 (with annually actualized versions in 2013 to 20 2 1) replaces the old model description originated back in 1997-1999 with numerous updates collected at the end of the document over the years (last version from November 2009). Thus, the actual description is internally much more consistent and offers a complete and detailed overview over the theory and the practical application of WaSiM.

The name of the model itself changed slightly: Since I didn't work at the ETH (Swiss Federal Institute of Technology or **E**idgenössische **T**echnische **H**ochschule in German) for many years now and since the model grew substantially in this time, I feel no longer entitled to decorate the name of the model with the abbreviation of that institution. So WaSiM may now be called simply WaSiM (Water Balance Simulation Model) – although another model with the same name but a different upper/lower case combination does exist. Also, when googling for WaSiM it is much easier to find references to WaSiM-ETH, so please feel free to further use the traditional name.

This document describes the theory of the model as well as its practical application. Configuration examples will be given throughout the text, but there are also some complete examples at the end of the manual. Some chapters of the old description found their way into this text, some didn't. However, I tried to keep the description as detailed as necessary without putting too much work into it. If there are any algorithms missing or not described in as much detail as needed, please contact me at j.schulla@wasim.ch . Also, there still exists a thorough description of the model theory in Schulla, J. (1997) for the topmodel version (sharing everything except the soil model related parts with the Richards-Version) which can be downloaded from the mentioned web site.

Almost all updates, descriptions, model versions and other material can be found at and downloaded from www.wasim.ch . I would like to thank Dr. Karsten Jasper for designing and maintaining the web site for several years (now, I'm trying to do that myself). WaSiM can be downloaded as executable version for Windows and Linux (32 bit and 64 bit). WaSiM still exists in two major versions. One is the version using the Topmodel-approach as runoff generation and soil model, the other is the version using the Richards approach as unsaturated zone and runoff generation model. Each of these versions is available for single processor and for multiprocessor systems using the OpenMP-standard, but not all are available for download for each patch or sub version – please contact me if you need a special compilation. An even more scalable version using Message Passing Interface (MPI) is available since February 2014 to be used on super computers for model configurations with a very large number of grid cells (should be compiled on each system to provide maximum performance and compatibility).

I wish all possible success to all those who already use or (still) wish to use the model.

Jörg Schulla, Zurich, Switzerland, June 2021 / updated May 2025

2 Model description

2.1 Basic principles and main features

WaSiM is a distributed, deterministic, mainly physically based hydrologic model. It runs in constant time steps (but nonetheless internally with very flexible sub time steps) on a regular or even irregular raster (the so-called model grid). For each time step, the sub models are processed one by one for the entire model grid thus taking most advantage of parallelized algorithms as offered by the OpenMP standard. Figure 2.1.1 shows the outline of the WaSiM model structure.

WaSiM main features are outlined in table 2.1.1.

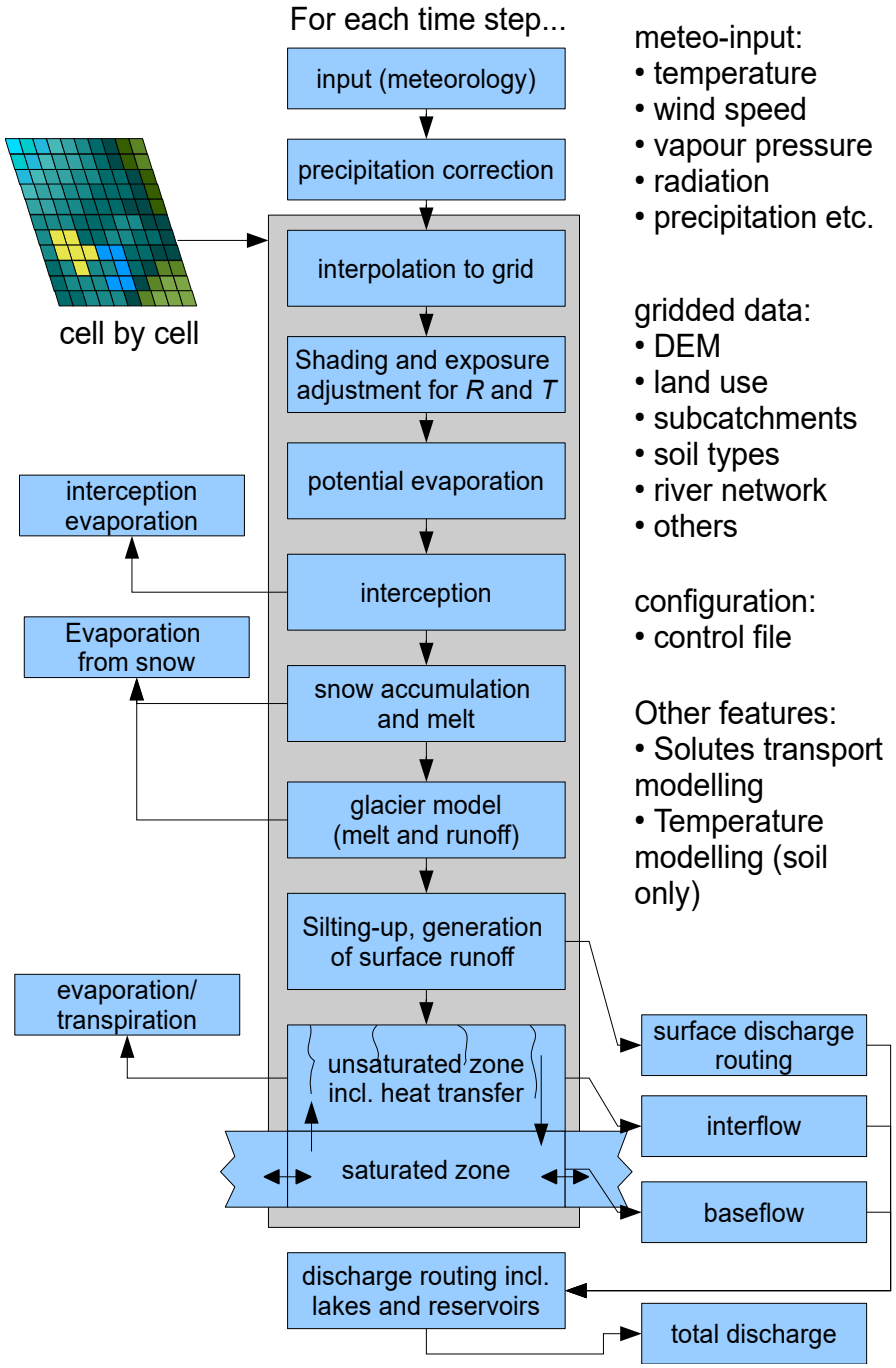


figure 2.1.1: model structure of WaSiM

Table 2.1.1: WaSiM main features

feature	comment/short description
Water balance related (basic features)	Not all of these features have to be used in a standard application, but these are the most common components
Interpolation of input data	Several methods can be used for down-scaling various meteorologic input data: - inverse distance weighting interpolation - elevation dependent regression (external and internal) - Thiessen polygons - bi-linear interpolation - bi-cubic splines - linear combination of bi-linear gradients and residuals - linear combination of bi-cubically splined gradients and residuals - reading externally interpolated data (as time series of grids) - Thiessen combined with a fix gradient (e.g. altitudinal lapse rate for temperatures)
Precipitation correction	Separate for liquid and solid precipitation, including wind dependent and wind independent corrections
Shadowing, exposition correction	Using suitable sub time steps, the effective radiation onto a cell can be estimated regarding the solar geometry as well as local topography (shading by other cells). Effects on local temperature can be regarded as well.
Potential and real evaporation and transpiration	Several methods may be used for calculating the potential evaporation and transpiration: - Penman-Monteith (recommended but requires mosz detailed input data) - method after Hamon - method after Wending - method after Haude
Snow accumulation and melt	Accumulation by threshold temperature, melt can be calculated by various approached: - T-Index (degree day method) - T-U-Index - combination approach after Anderson (with a extended version using observed vapor pressure) - refreezing and storage of melt water in the snow pack possible
Interception storage and evaporation	- Separate calculation of potential interception evaporation possible (using other resistances than that used for transpiration) - effective storage capacity depends on leaf area index and covered soil fraction as well as on crop/land use type

infiltration/soil water modelling	Topmodel approach: - soil water model uses some storages in addition to the original topmodel approach - additional interflow component possible - soil compartment for root accessible water Richards-approach: - parameterization of the soil into computational layers of <1cm up to 1m - any number of different soil horizons can be defined with different hydraulic properties each (each horizon is then subdivided into a given number of computational layers) - interflow generation at layer boundaries possible -
Groundwater flow	Multiple aquifers may be defined (although this should be better realized in an externally coupled groundwater model)
Runoff generation - baseflow	base flow generation by exfiltration into the channel network
Runoff generation - interflow	Interflow generation at soil layer boundaries possible when there is a slope angle > 0 and a drop in hydraulic conductivity. Translation to the sub-basins outlet is done using a flow time histogram (isochronic method)
Runoff generation – surface flow	Surface runoff is generated either by saturation from above (higher precipitation intensity than hydraulic conductivity) or from below (groundwater rises near the surface or even above the surface). The translation to the sub-basins outlet is done using another flow time histogram as for but parallel to the interflow
Runoff concentration (surface)	Alternatively to the runoff concentration using the isochronic method, one can use the surface runoff concentration model using the kinematic wave approach. This sub model can be configured to use extremely short sub time steps down to and below 1 second (recommended for backwater conditions and/or very fine spatial resolutions)
Runoff routing (channels)	Using the kinematic wave approach, the routing can be described for the combination of a main river bed routing and a flood plain routing. The channel routing also uses internal sub time steps (of some minutes usually in order to keep track of faster runoff packages)
Special features	used for special applications only – optional modules
Regional combination of different interpolations	For very large and meteorologic inhomogeneous regions, the various interpolation methods can be linearly combined in any fashion
irrigation	Using various methods, irrigation can be modelled for each land use in one of the following ways: - by defining dates for irrigation - by defining intervals for irrigation

	- by defining limits for soil moisture (in terms of suction) which mustn't be overstepped - by defining a ratio of real to potential evaporation which mustn't be overstepped
Soil model	
Clay layer	An additional clay layer may be considered
Artificial drainage	Acts like an extra interflow and will be defined per grid cell by the density of the drainage network
Re-infiltrating channel discharge	If routed runoff flows through cells with low groundwater level, water from the river may re-infiltrate into the groundwater (this feature is working best with MPI, since all river cells were processed in a special rank in their correct downstream order. OpenMP may result in slightly different values when comparing several identical runs.)
Ponding water	For each cell a maximum amount of ponding water may be defined. Exceeding water will become surface runoff. This feature should not be used when surface routing is active (surface routing calculates it's own ponding water if required by the DEM)
Macropore runoff	Acts as a bypass for infiltration into the soil. Water is brought directly into deeper soil layers and fills the soil bottom up.
Dynamic time step	Since the numerical solution of the Richards equation is a linear approximation only, the time step must be short enough for the Courant condition to hold at any time. Especially for longer time steps and/or for thinner soil layers with high hydraulic conductivity this can be achieved by using a dynamic sub time step algorithm. The minimum sub time step can be configured (to find a compromise between accuracy and computation time)
Glacier model	There is a static and a dynamic glacier model implemented into WaSiM. The dynamic model additionally models the firn layer and lets the glaciers shrink or grow according to their mass balance at the end of the configured glacier model year (e.g. at September 30th); Also, the glaciers are therefor not restricted to their own subbasin but may be part of any other subbasin as long as a glacier does not belong to multiple subbasins.
Applying (climate) scenarios	Climate scenarios given as average change (percentage, absolute value, factors etc.) may be applied to current weather data to simulate future climate. This is done after interpolation.
Tracer and solutes transport	The model version with Richards-approach offers the possibility to model at maximum 9 tracers together with the water fluxes. Tracers my be evaporating or non-evaporating, radioactive or non-radioactive tracers.

	Examples are salt and tritium.
Multi-layered vegetation	For detailed modelling of the soil-vegetation-atmosphere-transfer, the vegetation can be regarded as layered. The number of layers is unlimited, but usually one would use not more than three layers (e.g. crops, shrubs, trees)
Dynamic phenology	Vegetation parameters are usually defined for several fixed Julian days during a year. However, to be able to estimate the effects of changing climates, the phenology may also use dynamically defined sampling points. There are four different methods provided for different types of vegetation (trees, crops etc.)
Silting up model	For heavy rain events, the reduction of the soils infiltration capacity can be modelled by this component. Additional surface flow can be generated
Permafrost model	Although yet very simple, this model allows a change in the soil parameters when the soil is frozen. During snow free times, the soil thaws from above using a very simple time related equation.
Lake model (incl. dynamic sub time step)	Lakes can be fully integrated into the soil and groundwater model. If this extended option is used, there is a lateral exchange of water possible between groundwater and lakes as well as there is a vertical exchange possible between the soil and the lake. Each lake has a related volume-outflow table assigned defining the outflow dependent on lake volume and date. Since small lakes may change their volume during a flood event rapidly, the internal time step of the lake balancing model can be much smaller than the constant model time step (down to seconds)
Dynamic Abstraction control in routing model	For each routing channel, external and internal abstractions and inflows may be defined. Thus it is possible to model anthropogenic artifacts. Also, basins can be modelled only partly (e.g. without the upstream area of a gauge)
Sub time step control for groundwater model and unsaturated zone/surface routing	When using the model in a very high resolution (e.g. 1m by 1m) and/or for longer time steps than would be appropriate for the Courant-condition to hold, the complete surface/soil/groundwater compartment can be configured to run in a sub time step loop. Thus, the surface routing, the unsaturated zone model and the groundwater model are encapsulated in an extra loop, (regardless of the dynamic sub time steps for surface routing and unsaturated zone). Instabilities in the groundwater model can be avoided this way.
Soil stability index	The Topmodel-version provides now the possibility of a soil stability index taking into account some soil properties, topographic properties and the water table to estimate an index of probability of land slides.

Soil temperature model	Models the soil temperature using a physically based approach (1D-heat diffusion). Phase change is taken into account, thus enabling the model to be used in permafrost regions for thaw depth modelling etc. Parameter for hydraulic functions will be corrected for the temperature dependent phase change, thus e.g. the infiltration and vertical soil water movement as well as the groundwater flow will be affected by the soil temperatures.
Interface features	
Two-way-coupling (e.g. external groundwater models)	Some applications may require more detailed algorithms for some sub models. An example is a detailed groundwater model with several inhomogeneous aquifers. WaSiM provides the possibility to couple the model to other models using a file interface. WaSiM writes its states to disks and waits then until the extarnal model
two-way-coupling to a flood fore-casting system	WaSiM provides an interface to the Deltares flood early warning system FEWS. There were some internal changes made to support the coupling: - elevation dependent regression can be done now within WaSiM (no preprocessing required any more) - WaSiM writes an XML diagnostics file for interpretation by FEWS - some error or warning messages have been downgraded to information only The coupling needs a special adapter, called WaSiMFEWS
XML log file	This is generated automatically and may contain errors, warnings, info's and debug messages (depending on the command line parameters). It is thought to be sued by the FEWS system (see above) but can be used as a protocol as well also in standalone mode

2.2 Correction of Precipitation

The correction of precipitation is carried out separately for rain and snow using the wind speed as a parameter. The differentiation between rain and snow is done using a threshold temperature.

$$P_{corr} = P \cdot \left(a_l + b_l \cdot u_w \right) \qquad T \geq T_{thr}$$
$$P_{corr} = P \cdot \left(a_s + b_s \cdot u_w \right) \qquad T < T_{thr} \qquad (2.2.1)$$

with P observed precipitation [mm]
Pc_{orr} corrected precipitation [mm]
T_{thr} threshold temperature snow/rain [°C]
u_w wind speed [m/s]
a_l, b_l correction parameters for liquid precipitation [-]
a_s, b_s correction parameters for solid precipitation [-]

The following snippet shows, how the precipitation correction is configured in the control file

```
[precipitation_correction]
0     # 0=no correction, 1 = correction with one parameter set, 2 = specific parameters for each
region
0.0   # Snow-rain-temperature
1.05  # liquid:   b in:  y = p(ax + b) + 0.5% constant
0.04  # liquid:   a in:  y = p(ax + b) and 4% more per m/s
1.15  # Snow:     b in:  y = p(ax + b) + 15% constant
0.15  # Snow:     a in:  y = p(ax + b) and 15% more per m/s
```

precipitation correction if using inverse-distance-weighting interpolation of precipitation:

If using IDW for interpolating the precipitation data (see chapter 1.2) the station data are read in directly. These station data are then corrected using equation (2.2.1) by interpolating wind speed and temperature to the locations of the meteorological stations used for precipitation data. Both data are interpolated using the method specified in the control file for these variables.

precipitation correction if using altitude dependent regression for interpolation of precipitation:

If using altitude dependent regression for interpolating the precipitation, the real positions of the meteorological stations are unknown (only altitude profiles of the variables). In these cases the correction is done by correcting the gradients in a way, that in a first step all of the three altitudinal ranges are subdivided into 10 sub-ranges. For the middle of each of these sub-ranges the temperature and wind speed are interpolated using the appropriate method specified in the respective section of the control file. Then the correction equation (2.2.1) is applied to the precipitation value of the sub-range center. The corrected precipitations are then used to calculate new altitudinal gradients. Since the gradients of wind speed and temperature doesn't have to be linear nor parallel it is not possible to correct the gradients by shifting or rotating.

Precipitation correction when reading gridded data (method 9) or using regional superposition

Gridded data will always be corrected on a cell by cell basis using the local values of the already interpolated precipitation, wind and temperature data. Equation 2.2.1 uses P, u and T for each local cell value if at least one of the three entities uses interpolation method 9 (read in gridded data) or if superposition is used.

When superposition is used, i.e. interpolation is done for multiple regions with separate parameters, the precipitation correction can either be done with a single parameter set for a_l, b_l, a_s, b_s and T_{thr}, or the parameters can be provided for each region separately as shown here:

```
[precipitation_correction]
2     # 0=no correction, 1 = correction with one parameter set, 2 = specific parameters for each
region
# 1     2       3       4       5       region, the below parameters are used for
0.0     0.5     0.5     0       0       # Snow-rain-temperature
1.05    1.03    1.04    1.02    1.1     # liquid:   b in:  y = p(ax + b) + 0.5% constant
0.04    0.03    0.04    0.05    0.02    # liquid:   a in:  y = p(ax + b) and 4% more per m/s
1.15    1.10    1.07    1.12    1.10    # Snow:     b in:  y = p(ax + b) + 15% constant
0.15    0.10    0.08    0.14    0.11    # Snow:     a in:  y = p(ax + b) and 15% more per m/s
```

One special feature of correcting precipitation for regional superposition is, that although the corrections are done per region with differing parameters, the implementation makes sure that the resulting areal precipitation pattern is always smoothly connecting between the regions. This is done by the same weighted linear interpolation as the superposition itself. Additionally, the shift between snow and rain is also done smoothly, so there will be no abrupt changes between snow and rain when reaching the threshold temperature.

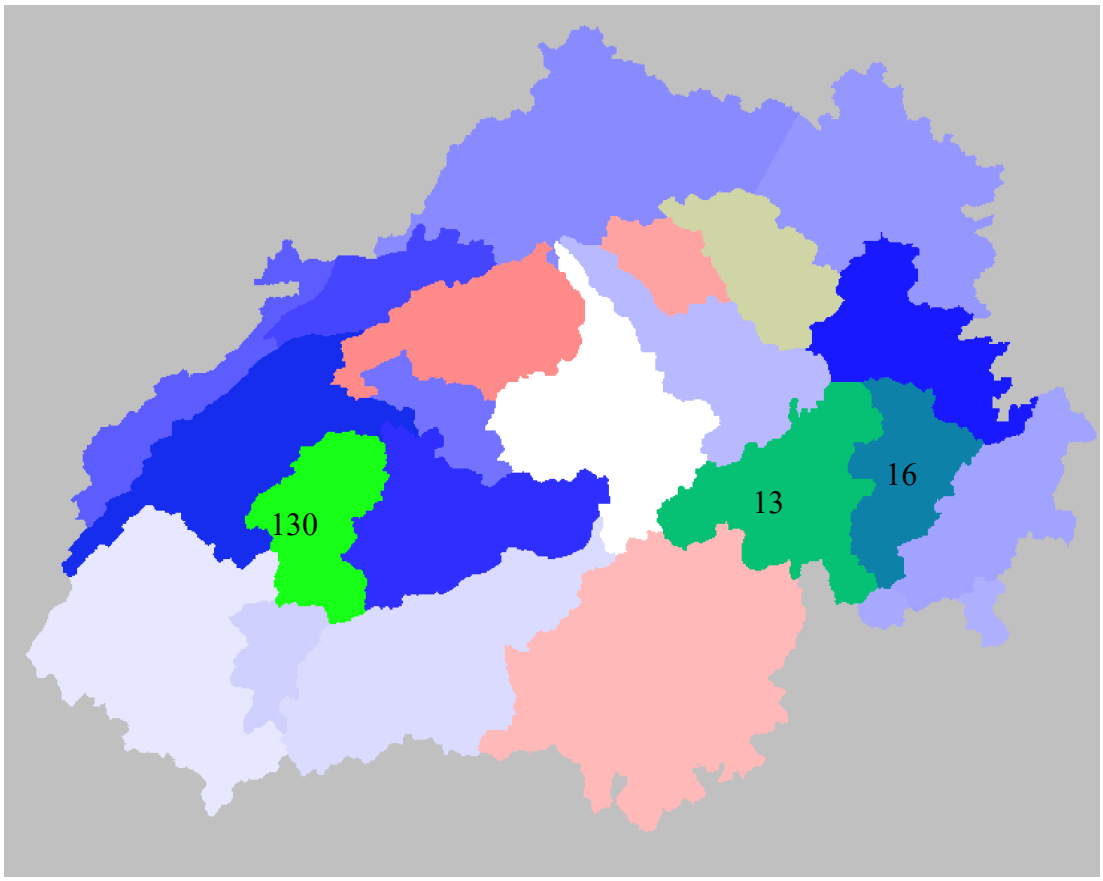


figure 2.2.1: Interpolation regions for Switzerland. The regions of interest for the examples are green and blue/green shaded with numbers 13, 16 and 130

An example of how the regional precipitation correction works is shown in figures 2.2.1 to 2.2.3. The regions in light green (regionb 130), green (13) and blue-green (16) are taken as examples for differing correction parameters.

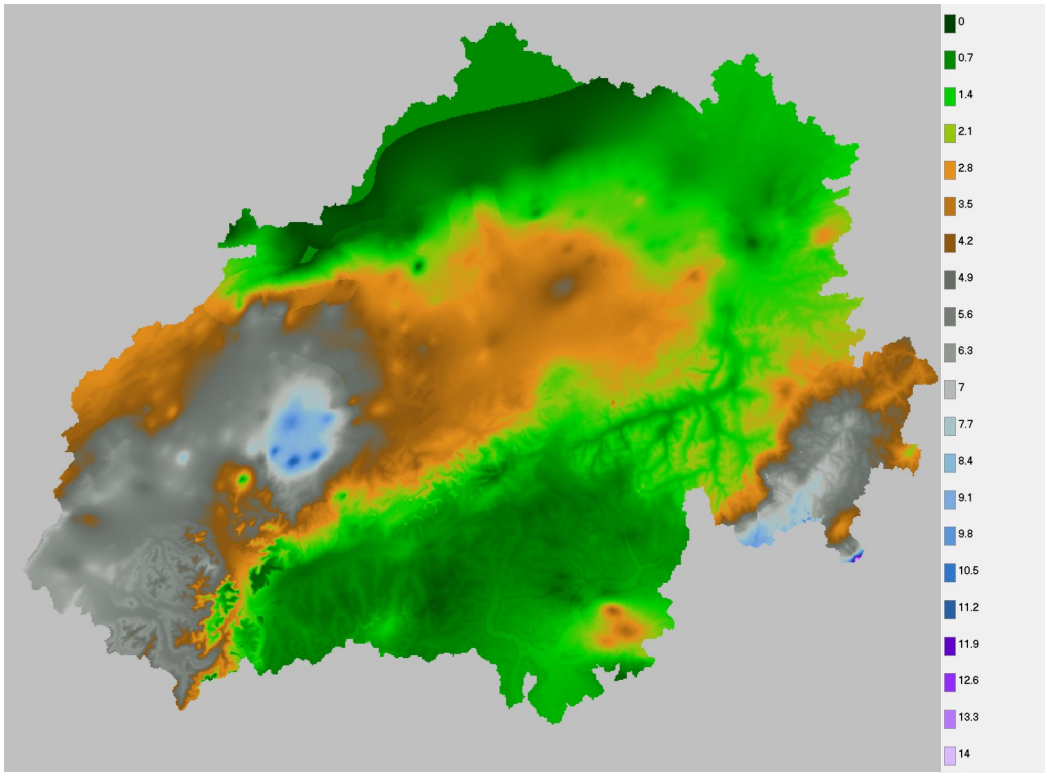


figure 2.2.2: interpolated precipitation with uniform precipitation correction for all regions

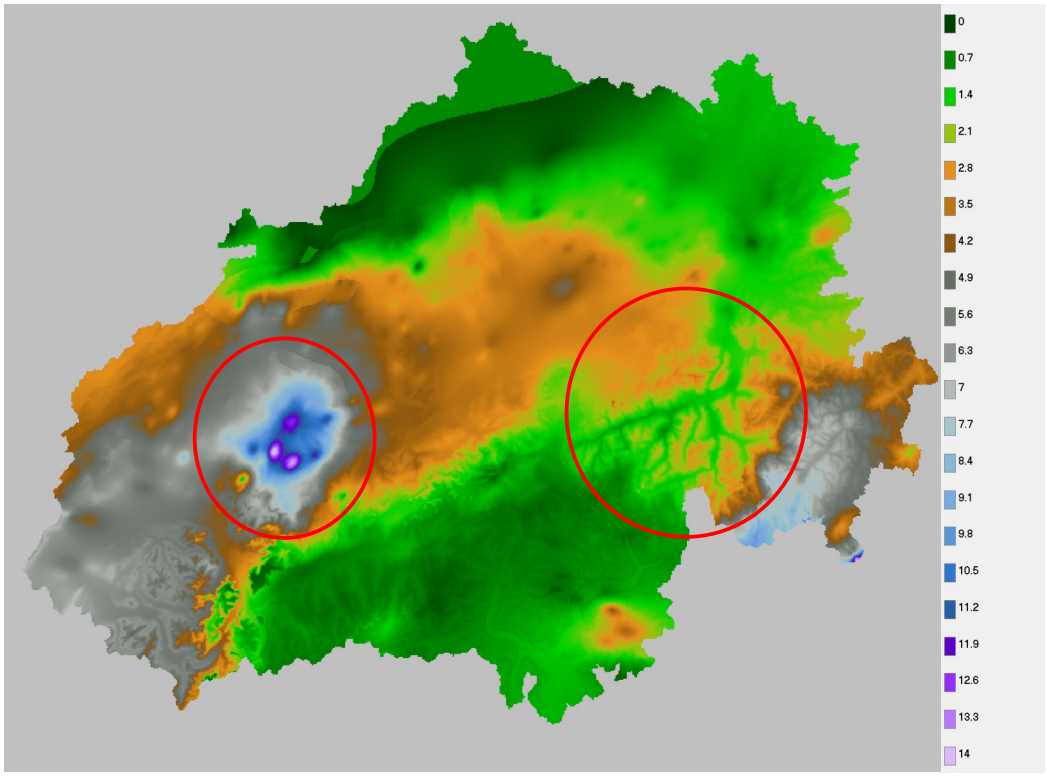


figure 2.2.3: interpolated precipitation with uniform precipitation correction for all regions

The parameters for precipitation correction were set as follows for the result in figure 2.2.2:

```
[precipitation_correction]
2  # 0=ignore this module, 1 = run the module, 2 = run with separate parameters per region
# 10    13      16      20      ...     110     120     130     ...# region
0.0     0.0     0.0     0.0     ...     0.0     0.0     0.0     ...# Snow-rain-temperature
1.05    1.05    1.05    1.05    ...     1.05    1.05    1.05    ...# liquid:   b in:  y = p(ax + b)
0.04    0.04    0.04    0.04    ...     0.04    0.04    0.04    ...# liquid:   a in:  y = p(ax + b)
1.15    1.15    1.15    1.15    ...     1.15    1.15    1.15    ...# Snow:     b in:  y = p(ax + b)
0.15    0.15    0.15    0.15    ...     0.15    0.15    0.15    ...# Snow:     a in:  y = p(ax + b)
```

All regions have the same parameter set, highlighted in red are the parameters sets for regions 13, 16 and 130 (not all regions appear here for clarity).

The corresponding parameter sets for the results shown in figure 2.2.3 are these :

```
[precipitation_correction]
2  # 0=ignore this module, 1 = run the module, 2 = run with separate parameters per region
# 10    13      16      20      ...     110     120     130     ...# region
0.0     0.0     0.0     0.0     ...     0.0     0.0     0.0     ...# Snow-rain-temperature
1.05    1.15    1.25    1.05    ...     1.05    1.05    1.15    ...# liquid:   b in:  y = p(ax + b)
0.04    0.14    0.24    0.04    ...     0.04    0.04    0.14    ...# liquid:   a in:  y = p(ax + b)
1.15    1.25    1.35    1.15    ...     1.15    1.15    1.25    ...# Snow:     b in:  y = p(ax + b)
0.15    0.25    0.35    0.15    ...     0.15    0.15    0.25    ...# Snow:     a in:  y = p(ax + b)
```

When comparing figures 2.2.2 and 2.2.3, it can be seen that

1) the precipitations in the marked regions 13, 16 and 130 are significantly higher in figure 2.2.3

2) there is no aprupt change but rather smooth transitions between the regions.

2.3 Interpolation of meteorological input data

WaSiM provides various methods to interpolate input data with a sparse spatial resolution to the model grid resolution. Usually, input data are given as time series of climate stations or as sparse input grids or pseudo station time series of climate models. The interpolation can be carried out by using one or more of the methods described below. As new feature starting with WaSiM 8.05.00 multiple interpolation methods may be linearly combined with different weights for different regions, thus offering a great flexibility.

The available methods are (numbers refer to the method-ID used in the control file):

1. Inverse Distance Interpolation (IDW). Use: for precipitation, sunshine duration and other data which are not (strongly) dependent on elevation or when modelling flat regions without substantial elevation ranges
2. Elevation Dependent Regression with External Pre-processing (EDREXT). Use: for all elevation dependent types of input data like temperatures, vapor pressure, air humidity, wind speed. Recommended for basins with a substantial elevation range only
3. A linear combination of IDW + EDREXT. Use: for data that depend only partly on elevation, e.g. precipitation in high mountain areas.
4. Thiessen Polygons. Use: if only one station is available (same results as IDW or EDR but faster)
5. Bi-linear Interpolation (BLI). Use: when using gridded time series from RCM or GCM output (WaSiM-tables containing time series for each grid cell of the GCM run)
6. Bi-linear Interpolation of both gradients and residuals and a linear combination thereof (BIGRES). Use: same as BLI but for data which are separately available as residues and gradients
7. Bi-cubic Spline Interpolation (BSI). Use: same as BLI but with another (smoother) technique
8. Bi-cubic Splice Interpolation of both gradients and residuals and a linear combination thereof (BSIRES). Use: same as BIGRES, but somewhat smoother
9. A pseudo-method: Reading externally interpolated data from grids; Use: when gridded input is already available (in the model raster dimensions!) from external interpolation routines
10. Elevation dependent regression with internal processing (EDRINT); like EDR but without external preprocessing using regr or regress. Use: same as method 2. The main difference between methods 2 and 10 is, that for method 2 the stations to be used can be selected manually in the ini file for the regress tool wheres for method 10 all stations in the input file will be used. Also, the internal preprocessing does not allow the mixing of input files with different temporally resolution as it is possible in regr and regress. To achieve identical results, the input file used for method 10 should only contain the stations which were used in the regress-ini-file, too.
11. A linear combination of IDW and EDRINT, like IDW+EDREXT but without external preprocessing using regr or regress. Use: same as for method 3; read comments on method 10 here above.
12. A combination of nearest neighbor (Thiessen) and a constant lapse rate

Each of these methods is described in detail in the following chapters.

2.3.1 Inverse Distance Interpolation (Method 1)

Method ID for all methods using this algorithm: 1, 3, and 11

For distance weighted interpolation there are two possibilities offered in WaSiM: the inverse distance weighting method and the Thiessen-polygon method, whereby Thiessen-polygons can be seen as a special case of IDW using only the nearest station for getting an “interpolation” result.

The input data are read in as a WaSiM-table. Regression gradient files cannot be processed since they contain no station data but only gradients. For the IDW-method all stations within a specified search radius are used for the interpolation. If all stations have identical values, the interpolation is skipped and the constant value is taken instead. If all stations have nodata values, the interpolation is skipped and the results from the last time step is taken instead.

The interpolation result is the sum of all contributing weighted station data:

$$\hat{z}(u) = \sum_j \left(w_j \cdot z(u_j) \right) \qquad \textbf{(2.3.1)}$$

$$\text{with} \quad w_j = \frac{1}{d(u,u_j)^p} \cdot \frac{1}{C} \quad \text{and} \quad C = \sum_j \frac{1}{d(u,u_j)^p} \quad \text{follows:} \quad \sum_j w_j = 1.0$$

with $\hat{z}(u)$ interpolated value at location u
w_j weight of the observed value at the station j
$z(u_j)$ observed value at the station j
$d(u,u_j)$ distance to the station j
p weighting power of the inverse distance (between 1 and 3, 2 is recommended)

WaSiM offers the possibility to change the parameter p, to specify a maximum distance dmax as well as two parameters specifying the anisotropy, which can be seen as an ellipse with a specified slope of the main axis against the horizontal (west-east-line, using mathematically positive rotation orientation) and with a specified length ratio of the long (main) axis to the short axis. The slope of the main axis may range from 0 to <90°, the axis length ratio may range from greater than zero to 1 (the latter would indicate that there is no anisotropy at all).

Practical application:

Control file: (red, bold and large font: specific settings for method IDW)

```
[precipitation]
1                          # method
$inpath//regen//$year//.dat AdditionalColumns=0   # station data
$inpath//regen//$year//.out  # regression data (if method = 2 or 3)
$outpath//idw_//$preci_grid  # name of the output grid
$Writegrid                   # writecode
0.1                          # correction faktor for results
$outpath//idw_prec//$grid//.//$year $hour_mean  # statistics file
998                          # error value
$IDWweight      # weighting of the reciprocal distance for IDW
0.75            # for method 3: relative weight of IDW
$IDWmaxdist     # max. distance of stations
$Anisoslope     #  slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie    #  ratio short to long axis of anisotropy-ellipsis
0.1               # lower limit of interpolation results
0                 # replace value for results below the lower limit
900               # upper limit for interpolation results
900               # replace value
$SzenUse    # 1=use scenario data for correction, 0=dont use scenarios
3       # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1       # number of scenario cells
699000 235000    0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

Examples:

Figures 2.3.1.1 and 2.3.1.2 show typical examples of pure IDW and pure EDR interpolations for the

same date (January 3, 1994, 17:00). To get the most information out of the data, a combination of IDW and EDR can be used → see chapter 1.2.3.

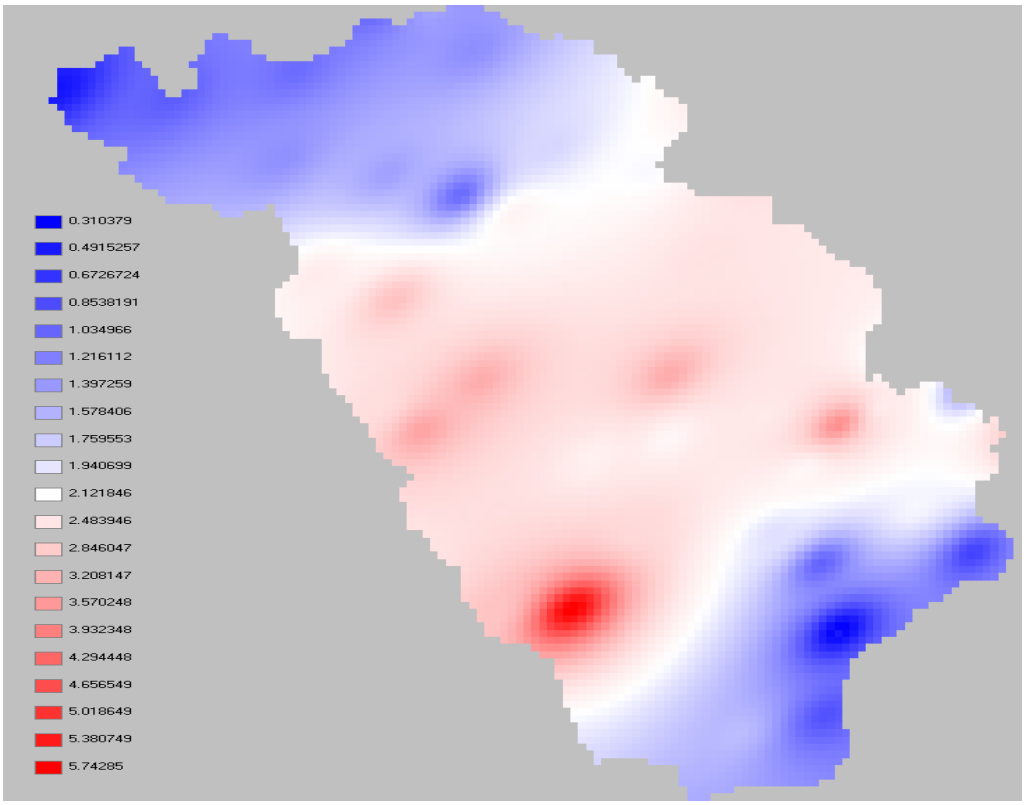


figure 2.3.1.1: example for IDW interpolation of precipitation (January 3, 1994, 18:00)

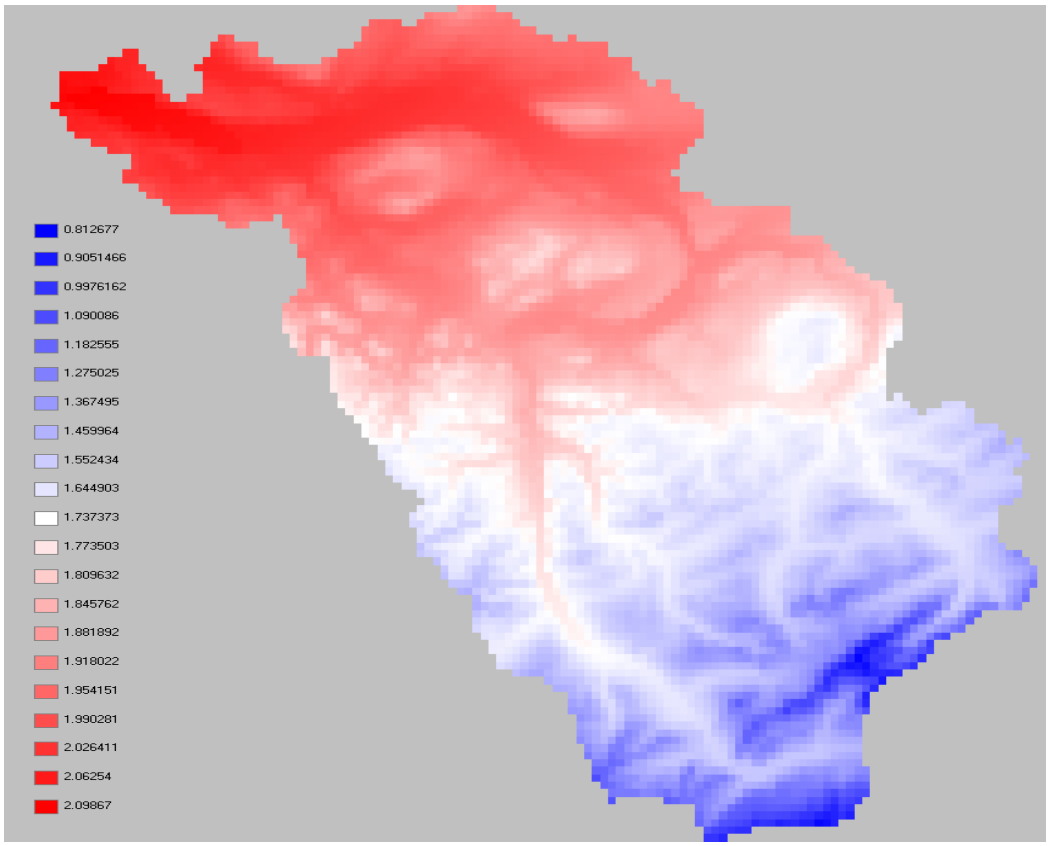


figure 2.3.1.2: example for EDR interpolation for the same data as in figure 2.3.1.1

2.3.2 Elevation dependent regression (Method 2)

Method ID for all methods using this algorithm: 2, 3, 10 and 11

Using this method, the interpolation result is estimated using the elevation dependence of a variable. If there are data of more than three station available, an additional trend surface is estimated in order to correct shifts in the horizontal directions. The trend surface uses only the residuals of the elevation dependent regression. To use this method within WaSiM it is either necessary to generate the parameter files containing the elevation gradients for each of these variables using the programs “regress” or “regr” or to use the internal regression methods.

The gradients will be estimated for at maximum three elevation ranges separated by inversions from each other. The estimation of the elevation of these inversions is done semi automatically by searching for intersections of the regressions of neighboring elevation ranges. As initial values typical elevation ranges of inversions has to be specified for each of the two possible inversions. Separate regressions are then estimated for the ranges below and above the lower inversion. If there is an intersection of both regressions between the lower and the upper (theoretical) inversion (plus an adjustable tolerance), then it is assumed that there is only one inversion at all. Otherwise, i.e. if the regressions are crossing outside of the range between the two inversions (plus tolerance), a twofold inversion is assumed making the lower regression valid below the lower inversion only and the upper regression above the upper inversion. For the elevation range between both inversions a third gradient is assumed to link both regressions linearly.

The interpolation to a specified spot in the model is then done by :

$$T(h_M) = a_i + b_i \cdot h_M \qquad \textbf{(2.3.2)}$$

with
- h_M elevation [m]
- T variable (e.g. air temperature)
- i index for lower, medium resp. upper regression (i=1..3)
- a_i, b_i parameter of the respective regression (i=1...3)

For the actual sample, parameters a and b are estimated as $\hat{a}$ and $\hat{b}$ using a least square fit (linear regression):

$$g(\hat{a}, \hat{b}) = \sum_i \left(T_i - \hat{a} - \hat{b} \cdot h_{M,i} \right)^2 \rightarrow Min \qquad \textbf{(2.3.3)}$$

The partial derivatives of (2.3.3) for $\hat{a}$ and $\hat{b}$ result in a linear equation system with $\hat{a}$ and $\hat{b}$ as the unknowns, which may be estimated by:

$$\hat{a} = \frac{1}{n} \cdot \left(\sum T_i - \hat{b} \sum h_{M,i} \right) \quad \text{and} \quad \hat{b} = \frac{n \cdot \sum T_i h_{M,i} - \sum T_i \sum h_{M,i}}{n \cdot \sum h_{M,i}^2 - \left(\sum h_{M,i} \right)^2} \qquad \textbf{(2.3.4)}$$

By applying equation (2.3.4) to all elevation data of the actual sample and calculating the difference between this regression value the actual value of each sample data, the sample's residues r_i with respect to the regression range are calculated:

$$r_i = \begin{cases} T_i - \left(a_3 + b_3 \cdot h_{M,i} \right) + a_1 & h_{M,i} > h_{v,u} \\ T_i - \left(a_2 + b_2 \cdot h_{M,i} \right) + a_1 \quad \text{for} & h_{v,l} < h_{M,i} \le h_{v,u} \\ T_i - \left(b_1 \cdot h_{M,i} \right) & h_{M,i} \le h_{v,l} \end{cases} \qquad \textbf{(2.3.5)}$$

with
- r_i residuum
- T_i observation value I (i=1...n; n=number of values)
- $a_{1...3}$ regression parameters $\hat{a}$ after (2.3.3) for the elevation ranges 1 to 3

$b_{1...3}$ regression parameters $\hat{b}$ after (2.3.3) for the elevation ranges 1 to 3
$h_{M,i}$ elevations of stations used for sample data
$h_{v,l}$ lower inversion (given as a parameter in m a.s.l.)
$h_{v,u}$ upper inversion (given as a parameter in m a.s.l.)

Note : For performance optimization, the value of a_1 is added to all r_i -values. Thus, the application of the regression parameters for the later interpolation is somewhat faster because most grid cells will be below the lower inversion usually (and by eliminating the offset a_1, two additions are saved)

To fit a trend surface through these residues, a multiple linear regression is carried out:

$$r = u + v_1 x + v_2 y \qquad \textbf{(2.3.6)}$$

with r fitted residuum for location x,y
u, v_1, v_2 regression parameters for the multiple linear regression
x,y coordinates of the station location with respect to the grid coordinates

By minimizing the square error, the parameters u, v_1 and v_2 are estimated as $\hat{u}$, $\hat{v}_1$ and $\hat{v}_2$:

$$\hat{v}_1 = \frac{1}{D} \cdot \begin{vmatrix} n & \sum r & \sum y \\ \sum x & \sum rx & \sum xy \\ \sum y & \sum ry & \sum y^2 \end{vmatrix} \qquad \hat{v}_2 = \frac{1}{D} \cdot \begin{vmatrix} n & \sum x & \sum r \\ \sum x & \sum x^2 & \sum rx \\ \sum y & \sum xy & \sum ry \end{vmatrix} \qquad \textbf{(2.3.7)}$$

with $$D = \begin{vmatrix} n & \sum x & \sum y \\ \sum x & \sum x^2 & \sum xy \\ \sum y & \sum xy & \sum y^2 \end{vmatrix} \quad \text{and} \quad \hat{u} = \bar{r} - \hat{v}_1 \bar{x} - \hat{v}_2 \bar{y} \qquad \textbf{(2.3.8)}$$

By using the parameter $\hat{a}_{1...3}, \hat{b}_{1...3}, \hat{u}, \hat{v}_1$ and $\hat{v}_2$ the interpolation can be carried for each elevation $h_{M,i}$ and for each set of coordinates x and y :

$$T_r = \begin{cases} \hat{u} + \hat{v}_1 x + \hat{v}_2 y - \hat{a}_1 + \hat{a}_3 + \hat{b}_3 z & z > h_{v,u} \\ \hat{u} + \hat{v}_1 x + \hat{v}_2 y - \hat{a}_1 + \hat{a}_2 + \hat{b}_2 z \quad \text{for} & h_{v,l} \le z \le h_{v,u} \\ \hat{u} + \hat{v}_1 x + \hat{v}_2 y + \hat{b}_1 z & z < h_{v,l} \end{cases} \qquad \textbf{(2.3.9)}$$

with T_r interpolated value
z elevation at coordinate x,y

Note : since all residues where extended by the value of $\hat{a}_1$, the multiple linear regression in equation (2.3.9) has to account for this offset by subtracting $\hat{a}_1$ from all regressions, resulting in a faster calculation for all grid cells below the lower inversion boundary $h_{v,l}$ (last row in (2.3.9).

Practical application

Control file:

the following example shows the settings for an interpolator using elevation dependent regression. There are two variants shows: first the traditional method using external preprocessing and second the new version with internal regression processing. The important parameters for elevation dependent regression are written in a red, bold, enlarged font. See also description of the control file in this manual for more information about the other parameters:

```
[temperature]
2  # method, here 2 for regression with external preprocessing
$inpath//tempe//$year//a.dat AdditionalColumns=0 # station data
$inpath//temper//$year//.out # file name with regression data
$outpath//reg_//$tempegrid   # name of the output grid
$Writegrid                   #
0.1                          # correction faktor for results
$outpath//reg_temp//$grid//.//$year $hour_mean # statistics file
998                          # error value
$IDWweight      # weighting of the reciprocal distance for IDW
0.1             # relative weight of IDW-interpolation
$IDWmaxdist     # max. distance of stations
$Anisoslope     # slope of mean axis of the anisotropy-ellipsis
$Anisotropie    #  ratio short to long axis of the anisotropy-ellipsis
-40                  #  lower limit of interpolation results
-40                  #  replace value for results below the lower limit
40                   #  upper limit for interpolation results
40             #   replace value for results with larger values than the upper
limit
$SzenUse    #  1=use scenario data for correction, 0=dont use scenarios
1           # 1=add scenarios, 2=multiply scenarios, 3=percentual change
4           #  number of scenario cells
...
```

and here the example for method 10 (and 11) with internal regression (pre-)processing

```
[temperature]
 10   # method, here 10 for regression with internal preprocessing
$inpath//tempe//$year//a.dat AdditionalColumns=0 # station data
820 1400 200 1 300 # lower inversion [m asl], upper inversion
# [m asl], tolerance [m], overlap [0/1 for true/false],
# clusterlimit [m]
$outpath//reg_//$tempegrid   # name of the output grid
$Writegrid                   #
0.1                          # correction faktor for results
$outpath//reg_temp//$grid//.//$year $hour_mean # statistics file
998                          # error value
$IDWweight      # weighting of the reciprocal distance for IDW
0.1             # relative weight of IDW-interpolation
$IDWmaxdist     # max. distance of stations
$Anisoslope     # slope of mean axis of the anisotropy-ellipsis
$Anisotropie    #  ratio short to long axis of the anisotropy-ellipsis
-40                  #  lower limit of interpolation results
-40                  #  replace value for results below the lower limit
40                   #  upper limit for interpolation results
40             #   replace value for results with larger values than the upper
limit
$SzenUse    #  1=use scenario data for correction, 0=dont use scenarios
1           # 1=add scenarios, 2=multiply scenarios, 3=percentual change
4           #  number of scenario cells
```

When using methods 10 or 11, there will be no name for an externally produced parameter file with regression coefficients. Instead, the most important parameters must be given there. Their meaning is as follows (compare description of regression programs “regr” and “regress”):

- 1 st parameter: lower inversion in m (above sea level)
- 2 nd parameter: upper inversion in m (above sea level)
- 3 rd parameter: tolerance (when the upper and the lower regression intersect within [$h_{v,l}$ -tolerance … $h_{v,u}$ +tolerance], then only two regressions are used, otherwise three

regressions will be used.

- 4 th parameter: boolean value, 0 for strictly separating both elevation ranges when estimating the regression parameters, 1 for allowing an overlap: the lowest station of the upper elevation range (above lower inversion) will be used for the lower regression and the highest station of the lower regression range (below lower inversion) will be used for the upper regression, thus reducing the risk of to sharp changes in gradients when the number of stations is relatively low in one or both elevation ranges
- 5 th parameter: minimum elevation spread for the stations in one elevation range to be really used for a linear regression – if all stations fall within this range (i.e. the difference in elevations of the highest station and the lowest station in one elevation range is less that this parameter), then all stations will be handled as a cluster (all station data of this range will be averaged → this means, that usually no real regression will be done but constant values for the entire elevation range will be used.

Examples:

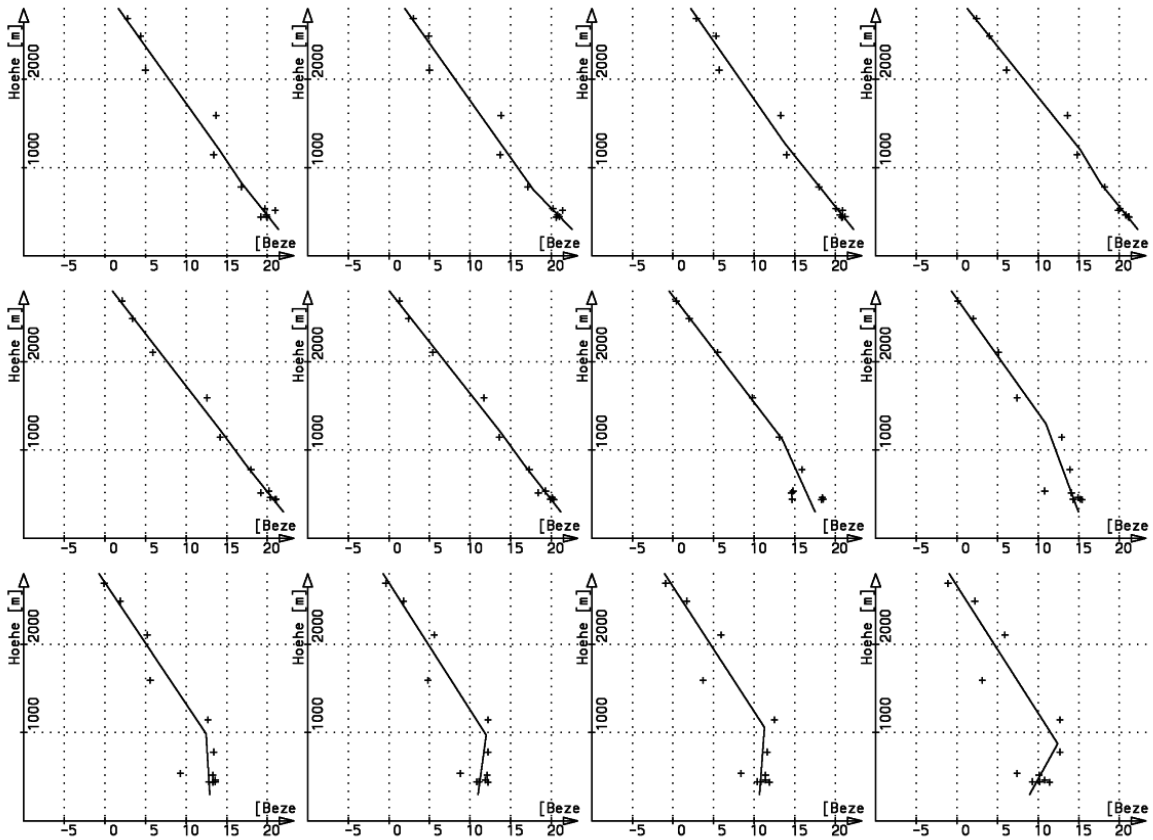


figure 2.3.2.1: typical temperature gradients (here for May 5, 1996, hours 1 to 12)

When using the tool regress, the elevations of the inversions can be estimated interactively (by trial and error). A typical screen output looks like the one shown in figure 2.3.2.1

In Figure 2.3.2.1, there are shown both cases: The upper right hand graph shows the temperature with three regression ranges because the uppermost and the lowermost regressions will not intersect between 820m and 1200m (although only barely visible here) whereas the lower right hand graph shows an example with only two active elevation ranges (because both regressions intersect between the upper and lower inversion.
The resulting lines in the preprocessing output file look like this (formatted with an additional header for convenience):

for the upper left hand example (measures in 1/10°C or m or (1/10°C)/m):

```
yy   mm dd hh   a1        b1        h v,l   a2        b2         hv,u    a3
1996  5  6  16  248.462  -0.08697  820.0  232.154  -0.06709  1200.0  255.908
 b3        u          v1                        v2
 -0.08688 145.71315 1.33328886195307E-0004 2.77000081244013E-0005
```

To calculate the temperature for a given cell with an elevation of 1000m and coordinates of x=731000 and y=248000, the second regression must be used (as well as parameter a_1 as described in the notes to equations 1.2.5 and 1.2.9). Using the parameters here, results in:

$$T = \hat{u} + \hat{v}_1 x + \hat{v}_2 y - \hat{a}_1 + \hat{a}_2 + \hat{b}_2 z$$
$$= 145.713 + 0.0001333 \cdot 731000 + 0.0000277 \cdot 248000 - 248.462 + 232.154 - 0.06709 \cdot 1000$$
$$= 166.62 = 16.62\,°C$$

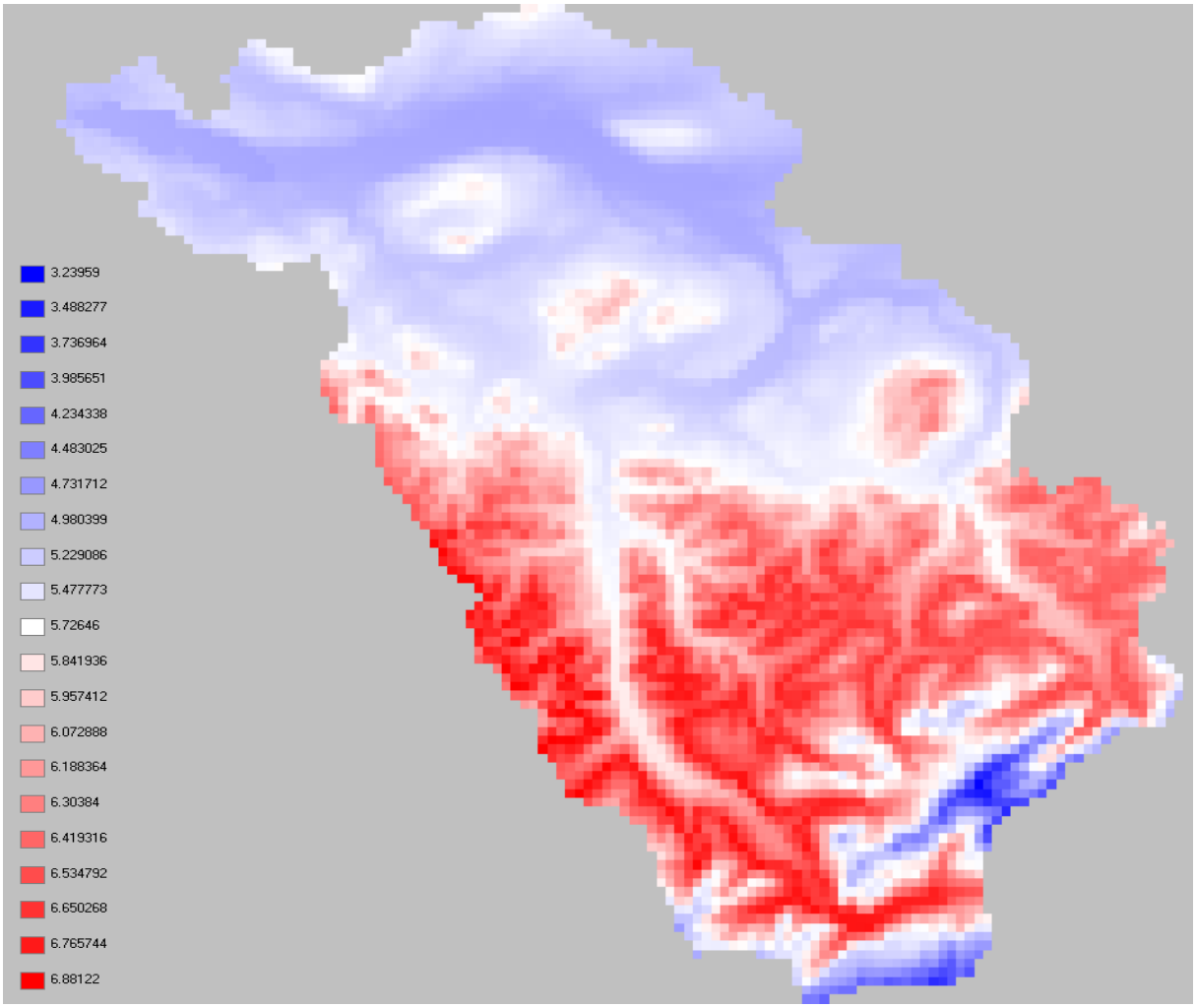


figure 2.3.2.2: Example for elevation dependent regression (January, 1, 1984, 14:00)

Figure 2.3.2.2 shows an example of elevation dependent regression for the first day of January 1998, hour 14. The high mountains in the South East show the deepest temperatures, the hilly regions have relatively high values and the "low lands" (~400...500m) in the North section have low values again: this is the typical pattern of a temperature inversion as shown in figure 2.3.2.1 in the lower left hand chart (but for another date). For the lower left hand graph in figure 2.3.2.1, the according parameter set looks like this:

```
yy   mm dd hh   a1      b1       h v,l  a2      b2       hv,u a3
1996 5  6  24   71.615 0.00000 0.0   71.615 0.06025 871.5 184.546
```

```
 b3         u          v1                        v2
-0.06933   220.05305 -2.3768683312431E-0004  9.86306324557048E-0005
```

Here, the lower inversion $h_{v,l}$ as well as the slope of the lowest regression b_1 is set to 0 to indicate, that only two regressions should be used. This mechanism may be problematic in regions with elevations < 0. The example above results in the following calculation for the interpolated value (same location as in the example before; here, the uppermost regression with index 3 is used):

$$T = \hat{u} + \hat{v}_1 x + \hat{v}_2 y - \hat{a}_1 + \hat{a}_3 + \hat{b}_3 z$$
$$= 220.053 - 0.0002377 \cdot 731000 + 0.0000986 \cdot 248000 - 71.615 + 184.546 - 0.06933 \cdot 1000$$
$$= 114.348 = 11.43\,°C$$

2.3.3 Linear combination of IDW + EDR (method 3)

Method ID for all methods using this algorithm: 3, and 11

This method combines methods 1 and 2 (using external preprocessing) and methods 1 and 10 (using internal regression preprocessing) respectively.

Practical application:

Control file: (red, bold and large font: specific settings for method IDW)

```
[precipitation]
3                          # method
$inpath//regen//$year//.dat AdditionalColumns=0   # station data
$inpath//regen//$year//.out  # regression data
$outpath//idw_//$preci_grid  # name of the output grid
$Writegrid                   # writecode
0.1                          # correction faktor for results
$outpath//idw_prec//$grid//.//$year $hour_mean  # statistics file
998                          # error value
$IDWweight      # weighting of the reciprocal distance for IDW
0.75            # for method 3: relative weight of IDW
$IDWmaxdist     # max. distance of stations
$Anisoslope     # slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie    # ratio short to long axis of anisotropy-ellipsis
0.1                 # lower limit of interpolation results
0                   # replace value for results below the lower limit
900                 # upper limit for interpolation results
900                 # replace value
$SzenUse    # 1=use scenario data for correction, 0=dont use scenarios


3       # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1       # number of scenario cells
699000 235000    0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

Examples:

Figure 2.3.3.1 shows a typical examples of a combination of IDW and EDR interpolations for the same date (January 3, 1994, 17:00) as used in the examples for IDW. The weights are 0.75 for IDW

and 0.25 for EDR.

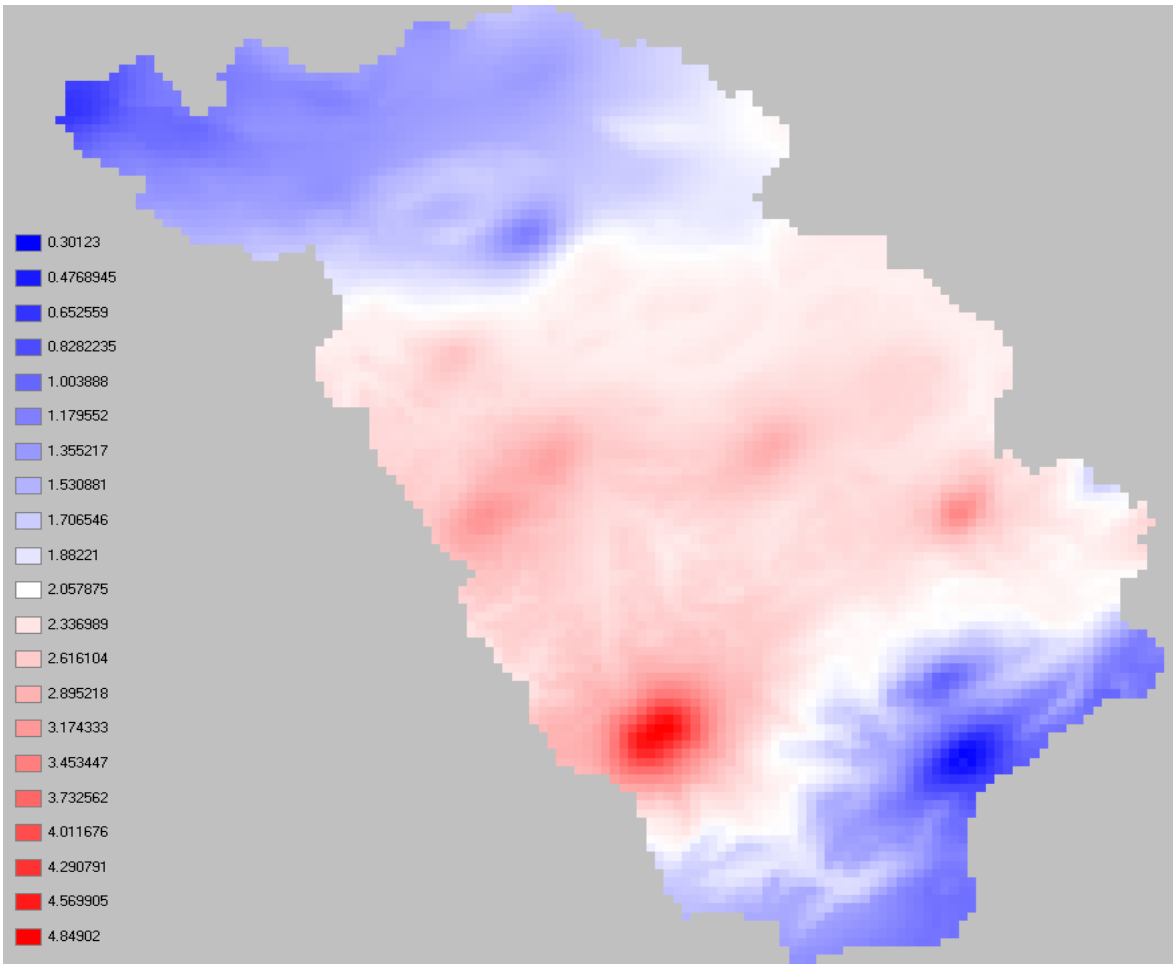


figure 2.3.3.1: example for a combination of IDW + EDR (January 3, 1984, 17:00), same data as for examples for method 1

2.3.4 Thiessen Polygons (method 4)

Thiessen polygons are the most simple method. It can be used if there is a single station only, but otherwise a more sophisticated method should be used. Each cell gets the value of the nearest station, thus creating a pattern of polygons delineating lines of equal distance to the two nearest stations.

Figure 2.3.4.1 shows an example for Thiessen polygons. The same data as for the examples in figures 2.3.3.1 and 2.3.2.2 are used.

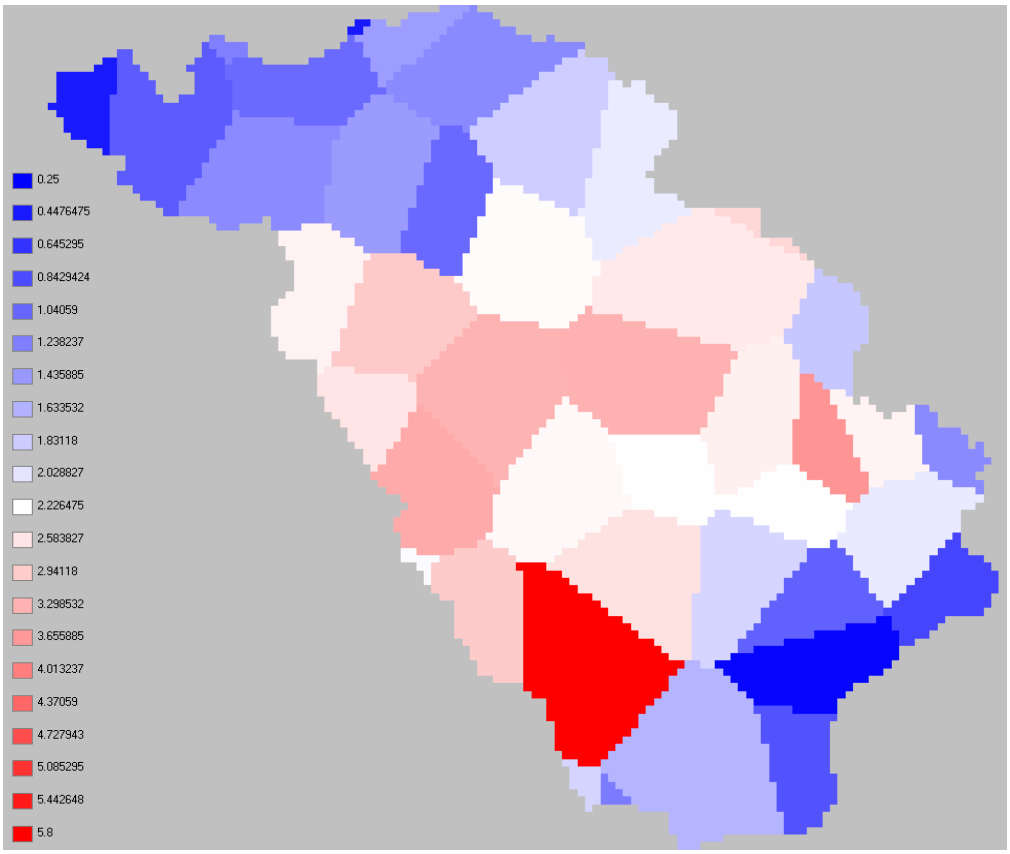


figure 2.3.4.1: example for interpolation using Thiessen polygones (precipitation, January 3, 1994, 17:00)

Practical application:

Control file: (red, bold and large font: specific settings for method IDW)

```
[precipitation]
4                                  # method
$inpath//regen//$year//.dat AdditionalColumns=0    # station data
$inpath//regen//$year//.out   # regression data
$outpath//idw_//$preci_grid   # name of the output grid
$Writegrid                    # writecode
0.1                           # correction faktor for results
$outpath//idw_prec//$grid//.//$year $hour_mean  # statistics file
998                           # error value
$IDWweight       # weighting of the reciprocal distance for IDW
0.75               # for method 3: relative weight of IDW
$IDWmaxdist      # max. distance of stations
$Anisoslope      #  slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie     #  ratio of short to long axis of the anisotropy-ellipsis
0.1                    # lower limit of interpolation results
0                      # replace value for results below the lower limit
900                    # upper limit for interpolation results
900                    # replace value
$SzenUse     # 1=use scenario data for correction, 0=dont use scenarios
3        # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1        # number of scenario cells
699000 235000    0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

2.3.5 Bi-linear interpolation (method 5)

Bi-linear interpolation is particularly useful for down scaling meteorological input data which are already gridded, e.g. in case of using outputs of global circulation models (GCM) or weather models to drive WaSiM. However, the gridded input data are not read in as grids but as usual station data files as described in chapter 3.2.2, representing the GCM grid cells by separate columns which contain the "station" coordinates (x, y, z) in the header and one date per row for each model time step. Bi-linear interpolation can also be used for usual climate station files, if the station positions are very close to a regular grid. Otherwise, artifacts may disturb a smooth interpolation surface.

Whilst usual bi-linear interpolation requires the input data in a regular grid with equal spaces between the grid nodes all over the grid covering the entire interpolation area, this constraint was slightly liberated in WaSiM: At initialization, for each grid cell the next input station is looked for in each of the four quadrants. The station indexes are stored in four internal index-grids to speed up the later interpolation by avoiding wasting time by searching the stations again and again in every time step. By looking for the four nearest stations in the respective quadrants, the stations does not have to be arranged in a grid order at all. However, the bi-linear interpolation assumes a more or less regular order, otherwise the interpolation results may show abrupt steps – but in principal it is possible to use any pattern of stations for this interpolation scheme. This liberation is also reasonable since GCM results are usually not given in regular grids but in latitude/longitude resolutions, so the distances between stations depend on the geographical location within the grid. If a station does not have four stations for interpolation available, i.e. if there is no station in at least one of the four quadrants or the actual data value is a nodata value, the interpolation method used for this cell is switched automatically to inverse distance weighting interpolation.

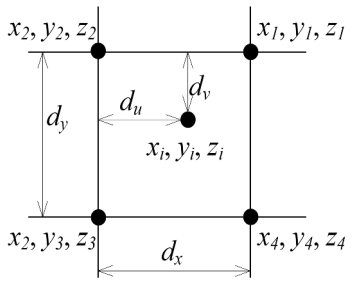


figure 2.3.5.1: scheme of calculating weights for bilinear interpolation to point i,j

Figure 2.3.5.1 shows the scheme of calculating the weights for the interpolation location i,j. The weights are given by:

$$u = \frac{d_u}{d_x} = \frac{x_i - x_2}{x_1 - x_2} \text{ and } v = \frac{d_v}{d_y} = \frac{y_i - y_2}{y_3 - y_2} \qquad \textbf{(2.3.10)}$$

with
- u weight of data of stations 1 and 4 (first and fourth quadrant) in x-direction
- v weight of data of stations 3 and 4 (third and fourth quadrant) in y-direction
- d_u distance of the interpolation location from x_2 in x-direction
- d_v distance of the interpolation location from y_2 in y-direction
- x_1, x_2 x-coordinates of „stations" 1 and 2 (in quadrants 1 and 2, resp.)
- y_2, y_3 y-coordinates of „stations" 2 and 3 (in quadrants 2 and 3, resp.)

The interpolation result is then calculated by:

$$z_{i,j} = u \cdot (1 - v) \cdot z_1 + (1 - u) \cdot (1 - v) \cdot z_2 + (1 - u) \cdot v \cdot z_3 + u \cdot v \cdot z_4 \qquad \textbf{(2.3.11)}$$

To tell WaSiM that bi-linear interpolation should be used, simply the interpolation method has to changed to 5 in the interpolation section of the respective climate variable.

Practical application:

Control file: (red, bold and large font: specific settings for method 5)

```
[precipitation]
5                              # method
$inpath//regen//$year//.dat AdditionalColumns=0   # station data
$inpath//regen//$year//.out  # regression data
$outpath//idw_//$preci_grid  # name of the output grid
$Writegrid                   # writecode
0.1                          # correction faktor for results
$outpath//idw_prec//$grid//.//$year $hour_mean  # statistics file
998                          # error value
$IDWweight      # weighting of the reciprocal distance for IDW
0.75              # for method 3: relative weight of IDW
$IDWmaxdist     # max. distance of stations
$Anisoslope     #  slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie    #  ratio of short to long axis of the anisotropy-ellipsis
0.1                  # lower limit of interpolation results
0                    # replace value for results below the lower limit
900                  # upper limit for interpolation results
900                  # replace value
$SzenUse    # 1=use scenario data for correction, 0=dont use scenarios
3       # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1       # number of scenario cells
699000 235000    0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

2.3.6 Linear combination of bi-linearly interpolated gradients and residuals (method 6)

When interpolating e.g. the temperature or other altitude dependent variables, it may be important to consider small scale local variations in the digital elevation model. Therefor, another interpolation was introduced into WaSiM, the so called **BIGRES**-method (Bi-linear **I**nterpolation of **G**radients and **RES**iduals). In fact, it is a linear combination of two independently performed bi-linear interpolations, the first one interpolating the gradients, the second one interpolating the residuals. Both input data have to be read in as usual "station"-files, i.e., the gradients and residuals have to be calculated externally.

The results of both interpolations are then linearly combined using the digital elevation model as scaling parameter for the gradients and the residuals as offset:

$$z_{i,j} = zr_{i,j} + h_{i,j} \cdot zg_{i,j} \qquad \textbf{(2.3.12)}$$

with
- $z_{i,j}$ result of the linear combination
- $zr_{i,j}$ result of the bilinear interpolation of the residuals (after 9.2.9)
- $zg_{i,j}$ result of the bilinear interpolation of the gradients (after 9.2.9)
- $h_{i,j}$ elevation from the digital elevation model [m]

Practical application:

Control file: (red, bold and large font: specific settings for method 6)

To use this interpolation method, the method parameter in the interpolation section has to be set to 6. The residuals are given as the first file (the usual station file name). Then, and this is important, the 2nd file name for the gradients follows as an additional file name. The file name(s) of the

regression file(s) must appear as usual! The following example shows a sample sequence from a control file using this interpolation method:

```
[temperature]
6                              # method
$inpath//t_0_100.dat AdditionalColumns=0    # station data
$inpath//grad100.dat AdditionalColumns=0    # station data
$inpath//temperature.out  # regression data
$outpath//tempgrid        # name of the output grid
$Writegrid                # writecode
0.1                       # correction faktor for results
$outpath//temp//$grid//.//$year $hour_mean  # statistics file
998                       # error value
$IDWweight      # weighting of the reciprocal distance for IDW
0.75              # for method 3: relative weight of IDW
$IDWmaxdist     # max. distance of stations
$Anisoslope     #  slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie    #  ratio of short to long axis of the anisotropy-ellipsis
0.1                  # lower limit of interpolation results
0                    # replace value for results below the lower limit
900                  # upper limit for interpolation results
900                  # replace value
$SzenUse    # 1=use scenario data for correction, 0=dont use scenarios
3       # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1       # number of scenario cells
699000 235000    0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

2.3.7 Bi-cubic spline interpolation (method 7)

As an extension to the bi-linear interpolation, also bi-cubic spline interpolation is applicable. The required input data are the same as for bi-linear interpolation. The interpolation is done in three steps:

1. The stations are automatically organized in a grid-like structure (like for bi-linear interpolation) including a virtual rotation of the domain
2. The north-south-meridians are spline-interpolated to the required density of information
3. An east-west-interpolation is done using the interpolated data of the meridians

Practical application:

Control file: (red, bold and large font: specific settings for method 7)

To use this interpolation-method, the control file must be manipulated as shown for bilinear interpolation, except that the interpolation code is 7.

```
[temperature]
7                              # method
$inpath//t_0_100.dat AdditionalColumns=0    # station data
$inpath//temperature.out  # regression data
$outpath//tempgrid        # name of the output grid
$Writegrid                # writecode
0.1                       # correction faktor for results
$outpath//temp//$grid//.//$year $hour_mean  # statistics file
998                       # error value
$IDWweight      # weighting of the reciprocal distance for IDW
0.75              # for method 3: relative weight of IDW
$IDWmaxdist     # max. distance of stations
$Anisoslope     #  slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie    #  ratio of short to long axis of the anisotropy-ellipsis
0.1                  # lower limit of interpolation results
```

```
0                    # replace value for results below the lower limit
900                  # upper limit for interpolation results
900                  # replace value
$SzenUse    # 1=use scenario data for correction, 0=dont use scenarios
3       # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1       # number of scenario cells
699000 235000    0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

2.3.8 Linear combination of spline interpolated gradients and residuals (method 8)

It's also possible to use a combination of bi-cubic-spline-interpolated residuals and gradients, like for bi-linear interpolation. The application is also identical like those, except that the interpolation code is 8 instead of 6.

The application of bi-cubic spline interpolation generally gives smoother surfaces, but it may happen, that unexpected results are shown: Since the interpolation makes a smooth curve ist possible, that minimum or maximum values exceed the value ranges of an entity. In this case the maximum and minimum parameter has to be set to proper values (e.g. for precipitation the minimum has to be set to 0, for humidity the minimum has to be set to 0, the maximum to 1).

Practical application:

Control file: (red, bold and large font: specific settings for method 8)

To use this interpolation method, the method parameter in the interpolation section has to be set to 6. The residuals are given as the first file (the usual station file name). Then, and this is important, the 2nd file name for the gradients follows as an additional file name. The file name(s) of the regression file(s) must appear as usual! The following example shows a sample sequence from a control file using this interpolation method:

```
[temperature]
8                          # method: combination of splines for
# residuals and gradients
$inpath//t_0_100.dat AdditionalColumns=0   # station data
$inpath//grad100.dat AdditionalColumns=0   # station data
$inpath//temperature.out  # regression data
$outpath//tempgrid        # name of the output grid
$Writegrid                # writecode
0.1                       # correction faktor for results
$outpath//temp//$grid//.//$year $hour_mean  # statistics file
998                       # error value
$IDWweight      # weighting of the reciprocal distance for IDW
0.75              # for method 3: relative weight of IDW
$IDWmaxdist     # max. distance of stations
$Anisoslope     #  slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie    #  ratio of short to long axis of the anisotropy-ellipsis
0.1                  # lower limit of interpolation results
0                    # replace value for results below the lower limit
900                  # upper limit for interpolation results
900                  # replace value
$SzenUse    # 1=use scenario data for correction, 0=dont use scenarios
3       # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1       # number of scenario cells
699000 235000    0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

2.3.9 Reading externally interpolated data as ASCII-grids, binary grids, GIF files or netCDF files (method 9)

If the input data are already available as interpolated grids, it is possible to configure WaSiM in a way that for each time step one of these grids is read instead of interpolating the data directly. Thus, outputs from GCM or RCM can be used with external interpolation methods. The only requirement is, that each grid is compatible to the grid dimensions and coordinates of the actual model run (ncols, nrows, cellsize, xll_corner and yll_corner). The grids may be organized on the hard disk or in any directory structure in any way – each grid may be specified with its full path and name.

If these grids are provided as GIF (e.g. Swiss radar precipitation data) or as netCDF files, the grid geometry may even differ from the internal model grid. In bilinear interpolation is then done under certain conditions (see below).

Practical application:

Control file: (red, bold and large font: specific settings for method 9 – reading externally provided grids)

```
[precipitation]
9                              # method
$inpath//precip_gridlist.dat AdditionalColumns=0 # WaSiM grid list
$inpath//precip//$year//.out  # regression data
$outpath//idw_//$preci_grid  # name of the output grid
$Writegrid                    # writecode
0.1                           # correction faktor for results
$outpath//idw_prec//$grid//.//$year $hour_mean  # statistics file
998                           # error value
$IDWweight      # weighting of the reciprocal distance for IDW
0.75              # for method 3: relative weight of IDW
$IDWmaxdist     # max. distance of stations
$Anisoslope     #  slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie    #  ratio of short to long axis of the anisotropy-ellipsis
0.1                  # lower limit of interpolation results
0                    # replace value for results below the lower limit
900                  # upper limit for interpolation results
900                  # replace value
$SzenUse    # 1=use scenario data for correction, 0=dont use scenarios
3       # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1       # number of scenario cells
699000 235000    0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

The format of the grid list file looks similar to the WaSiM table file, except that the (single) data column contains valid file names instead of station data. This is an example for the grid list file:

```
grid list preipitation
YY    MM    DD    HH    9999
YY    MM    DD    HH    9999
YY    MM    DD    HH    9999
YY    MM    DD    HH    Liste
1984  1     1     1     d:\Data\m500\input\meteogrids\precm500_0101.001
1984  1     1     2     d:\Data\m500\input\meteogrids\precm500_0101.002
1984  1     1     3     d:\Data\m500\input\meteogrids\precm500_0101.003
1984  1     1     4     d:\Data\m500\input\meteogrids\precm500_0101.004
1984  1     1     5     d:\Data\m500\input\meteogrids\precm500_0101.005
1984  1     1     6     d:\Data\m500\input\meteogrids\precm500_0101.006
...
```

Note: The delimiters should be tabulators, not spaces.

Using GIF files:

GIF files contain compressed integer values ranging from 0 to 255. The GIF format was originally invented for efficient network transfer of losslessly compressed image data and was a proprietary format owned and copyrighted by Compuserve and UniSys (the latter company owned the patent on the LZW compression used internally), so only after the patent protection period ended, free use of GIF was possible. To use GIF in WaSiMs interpolation method 9, a series of GIF files has to be prepared with a single file for each time step similar to reading ASCII- or binary WaSiM-grids as ahown above. In addition, the value-mapping must be defined. This is done by 5 additional parameters in the third line of the section, whereof the first two parameters describe an exponential function and the last two describe another exponential function, crossing at the abscissa given by the third parameter (here 127). It is recommended to use the default values shown below.

```
[precipitation]
9     # method: 9 = reading and interpolating externally provied grids
$inpath_meteo//CPC.lst AdditionalColumns=0 # file name with grid names
8.9167e-5 0.07675 127 8.9167e-5 0.07675 # --> normal Mapping; for GIF-Mapping:
# parameters f1, x1, threshold, f2 and x2 for two logarithmic regressions, the
# first one is valid for GIF values below and equal to threshold, the second
# set for GIF values above threshold (P=f1*exp(x1*GIFvalue) resp.
# P=f2*exp(x2*GIFvalue))
$outpath//$preci_grid     # name of the output grid
1//$Writegrid             # writecode
0.9                       # correction factor
$outpath//prec//$grid//.//$code//$year $hour_mean  # statistics file
998                       # error value
$IDWweight                # weighting of the reciprocal distance for IDW
0.75              # relative weight of IDW-interpolation in the result
$IDWmaxdist       # max. distance of stations to the actual interpolation cell
$Anisoslope           # slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie          # ratio of short to long axis of the anisotropy-ellipsis
0.1                   # lower limit of interpolation results
0                     # replace value for results below the lower limit
90                    # upper limit for interpolation results
90                    # replace value for results above the upper limit
$SzenUse          # 1=use scenario data for correction, 0=dont use scenarios
3                 # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1                 # number of scenario cells
699000 235000     0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

Sice the gif files (at least the ones from Swiss Meteo radar stations) cover always an area much larger than the modeled basin, an internal interpolation is required to extract data for the actual model region only. Therefore, some additional information has to be provided in the list file header.
- cellsize: must be given in the second header line
- xll_corner coordinate of the gif-data must be given in the third header row
- yll_corner coordinate of the gif-data must be given in the fourth header row

```
Radardata_Switzerland_2005
YY    MM    DD    HH    1000          --> cell size in m
YY    MM    DD    HH    255000        --> xll_corner coordinate
YY    MM    DD    HH    -160000       --> yll_corner coordinate
YY    MM    DD    HH    gif-list
2005  1     1     1     d:\Data\CPC\2005\CPC0500100000_00060.801.gif
2005  1     1     2     d:\Data\CPC\2005\CPC0500101002_00060.801.gif
2005  1     1     3     d:\Data\CPC\2005\CPC0500102002_00060.801.gif
2005  1     1     4     d:\Data\CPC\2005\CPC0500103001_00060.801.gif
2005  1     1     5     d:\Data\CPC\2005\CPC0500104001_00060.801.gif
...
```

A typical GIF-Image looks like the example in figure 2.3.9.1. The region of interest is marked with a rectangle. This original GIF has a resolution of 1000m. Figure 2.3.9.2 shows the interpolated model grid with a 500m resolution.

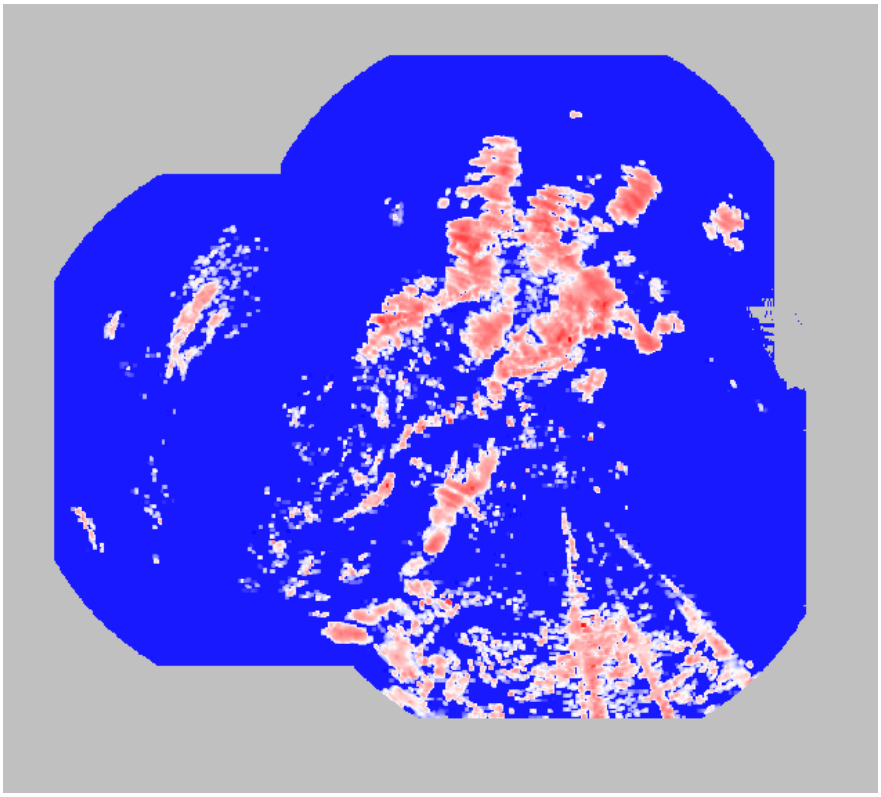


figure 2.3.9.1: Example for a Swiss radar data image covering entire Switzerland with three radar stations (unmapped values)

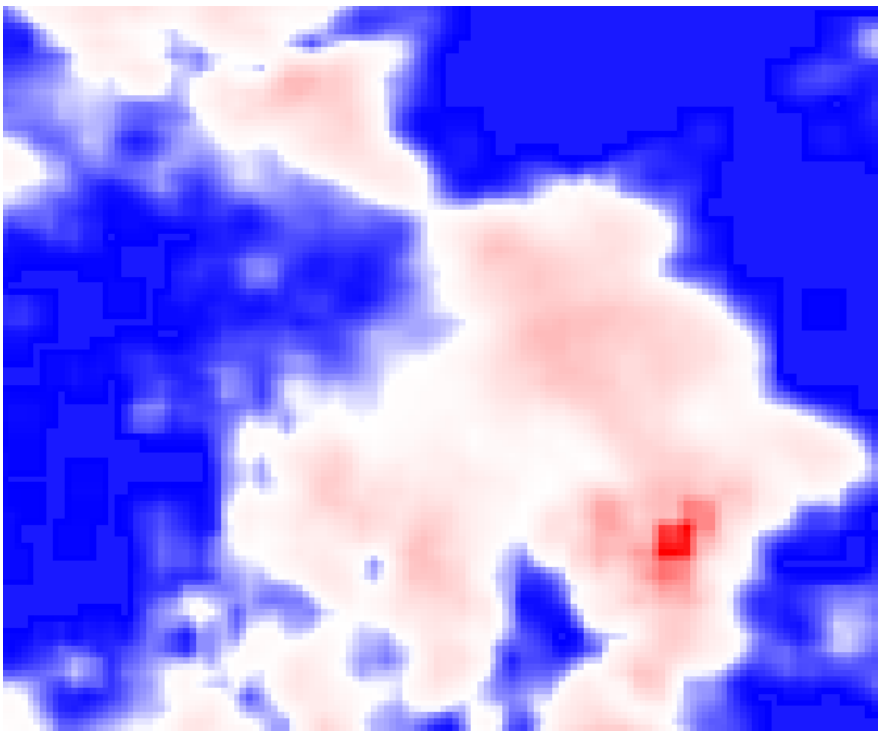


figure 2.3.9.2: interpolated region of interest, mapped to precipitation values

Radar data often show systematic errors. As a simple way of compensating for systematic errors (caused by shadowing or distance effects), WaSiM allows to specify a grid containing correction factors for each grid cell. This grid has to be read in as standard grid. The internal identifyer is "ScalingGrid_<entity>", e.g. "ScalingGrid_precipitation" when the interpolation entity is named precipitation. Each interpolator looks for such a grid. If found, the correction factor is applied on a cell-by-cell to the mapped GIF-values (can also be used for other interpolation methods).

Figure 2.3.9.3 shows such a correction factor grid. This grid can be obtained by running the interpolation of station data with any other method for a year or so and then running the same period with GIF-input data. The sum-grids can then be divided (i.e. (radar data) / (station data) = correction factor). Such a simple correction factor makes sure that the average precipitation for the above interpolated period is the same in both methods, however, it can result in quite different results when applied for other periods.

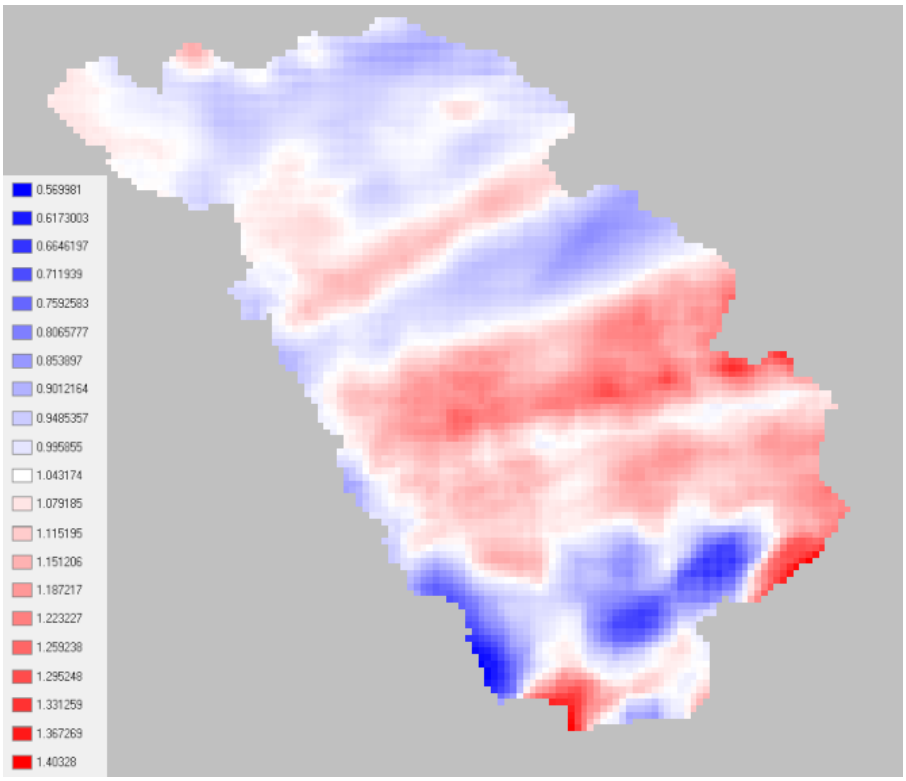


figure 2.3.9.3: Sample correction factor grid for radar data for the river Thur basin

Using netCDF files

The netCDF Format is a format for exchanging scientific rasterized data. Many GCMs use this format as output format for time series of e.g. precipitation fields. In order to simplify the application of WaSiM with such GCM output, the netCDF format was integrated into WaSiM with version 10.02.02. It can be used for reading in grids and for writing grids and stacks. Two properties of the netCDF format make it an interesting choice for output of time series of gridded data:

- netCDF allows compression

- netCDF allows saving a complete series of multi-dimensional data sets within one single file. Such, the write option 1 for grids available in WaSiM from day one becomes a completely new meaning: it is now possible to write a single netCDF grid containing the output grids of all timesteps, e.g. the complete series of interpolated precipitation or the time series of snow water equivalents etc.

There are a few pre-requisits for using netCDF format:

1. the appropriate netCDF library has to be installed on the system with its binary directory in the systems path. As an alternative, the 32bit or 64bit library package that is available on the WaSiM home page may be downloaded and installed (unpacked) in the WaSiM installation folder. There are a few DLLs (dynamic link libraries) that are required to read and write netCDF files (hdf5.dll, hdf5_hl.dll, netcdf.dll, zlib1.dll). On Linux systems, a netCDF package must be installed as well (no libs are provided on the WaSiM home page for Linux since each distribution may have their own file versions required).
2. For writing netCDF, use the new (optional) parameter WriteNetcdfGrids = 1 in the controlfile section [output_interval]. It can be combined with WriteDoubleGrids=1 in order to write netCDF in double preciosion or WriteDoubleGrids=0 in order to write netCDF in single precision.
3. For reading meteorologic files: Prepare the input list file as for reading other grids, but add for each entry the variable and the internal layer as follows:

   filename<variable name>layer, see example below.

```
grid list for pr_bav-EUR-11_observations_historical_NA_LfU-KliRef2014_v1_day
YY    MM    DD    HH    12132
YY    MM    DD    HH    463951
YY    MM    DD    HH    5123983
YY    MM    DD    HH    List-of-grid-files
1970  1     1     0     ORGNETCDF\pr_observations_19700101-19701231.nc<pr>0
1970  1     2     0     ORGNETCDF\pr_observations_19700101-19701231.nc<pr>1
1970  1     3     0     ORGNETCDF\pr_observations_19700101-19701231.nc<pr>2
1970  1     4     0     ORGNETCDF\pr_observations_19700101-19701231.nc<pr>3
```

The control file can be prepared like for the standard interpolation method 9, no mapping values must be given for netCDF inputs. However, as can be seen by the example above, the cellsize, xll_corner and yll_corner can be defined (optionally) in the list file. The processing follows this rules:

- if there are no cellsize, xll_corner and yll_corner coordinates given (i.e. all values are 9999), WaSiM assumes the netCDF file to contain grids with the same grid geometry than the internal grids, i.e. matching number of columns and rows, same cell size and same location with respect to the origin of the axes (xll, yll). If the number of rows and/or columns does not equal the internal grids, an error is thrown. Otherwise, the read in values are mapped 1:1 to the internal interpolation result. Note that WaSiM allows the netCDF data to define the

attributes scale_factor and add_offset which will be searched for in every netCDF input. If found, the add_offset value is added to each read in cell value after the scale_factor was applied to it as a factor (i.e. interpolation result = raw_value*scale_factor+ add_offset)

- if one of the values cellsize, xll_corner or yll_corner in the list file header is not 9999, then the values read from the netCDF file (with already applied scale_factor and add_offset) are interpolated to the WaSiM model grid. Here, the same algorithm that was used for GIF data is applied (bilinear interpolation between the four neighboring points in the read in data).

A netCDF file can be converted to it's ASCII representation by using the ncdump utility that is provided with the netCDF installation. A netCDF file (processable by WaSiM) in it's ASCII representation can look like this example:

```
netcdf pr.grd {
dimensions:
      x = 34 ;
      y = 37 ;
      t = UNLIMITED ; // (31 currently)
variables:
      double x(x) ;
            x:units = "m East of reference point" ;
      double y(y) ;
            y:units = "m North of reference point" ;
      double t(t) ;
      float pr(t, x, y) ;
            pr:missing_value = -9999. ;
data:

 x = 463951, 476083, 488215, 500347, 512479, 524611, 536743, 548875, 561007,
    573139, 585271, 597403, 609535, 621667, 633799, 645931, 658063, 670195,
    682327, 694459, 706591, 718723, 730855, 742987, 755119, 767251, 779383,
    791515, 803647, 815779, 827911, 840043, 852175, 864307 ;

 y = 5560735, 5548603, 5536471, 5524339, 5512207, 5500075, 5487943, 5475811,
    5463679, 5451547, 5439415, 5427283, 5415151, 5403019, 5390887, 5378755,
    5366623, 5354491, 5342359, 5330227, 5318095, 5305963, 5293831, 5281699,
    5269567, 5257435, 5245303, 5233171, 5221039, 5208907, 5196775, 5184643,
    5172511, 5160379, 5148247, 5136115, 5123983 ;

 t = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
    20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 ;

 pr =
  -9999, 0.117566, 0.1027475, 0.1179458, 0.3082994, 0.37783, 0.4329825,
    0.432856, 0.3330186, 0.3513955, 0.307881, -9999, -9999, -9999, -9999,
    0.3503753, 0.2788355, 0.247333, 0.2531627, 0.2667902, 0.3186074,
    0.3818424, 0.4888197, 0.6546389, 0.865576, 0.9123307, 0.9530244,
    1.078685, 1.34956, 1.666996, 1.167753, 1.222527, 1.562451, 1.677706,
    -9999, -9999, -9999,
  0.2858468, 0.478316, 0.6115415, 0.3465108, 0.4796911, 0.4885211, 0.3828494,
    0.3253893, 0.4304737, 0.2534156, 0.2371101, 0.2488278, 0.2402503,
    0.2215725, 0.3146775, 0.3095353, 0.247461, 0.2258213, 0.2674081,
    0.2897584, 0.3567264, 0.460595, 0.5657688, 0.6808999, 0.8345525,
    0.8813545, 0.8701995, 0.9644287, 1.156348, 1.321213, 1.073389, 1.314415,
    1.822002, -9999, -9999, -9999, -9999,
…
```

More information on how netCDF files are written by WaSiM are available in chapter 4.9.2 for writing and 4.6.1 for reading gridded data.

2.3.10 Elevation dependent Regression with internal preprocessing (method 10)

This method (EDRINT) was described in details in chapter 2.3.2 already. This is a reference to this section only.

Practical application:

Control file: (red, bold and large font: specific settings for method IDW)

```
[precipitation]
10                              # method
$inpath//regen//$year//.dat AdditionalColumns=0   # station data
820 1400 200 1 300 # lower inversion [m asl], upper inversion
#   [m   asl],   tolerance   [m],   overlap   [0/1   for   true/false],
#clusterlimit [m]
$inpath//regen//$year//.out  # regression data
$outpath//idw_//$preci_grid  # name of the output grid
$Writegrid                   # writecode
0.1                          # correction faktor for results
$outpath//idw_prec//$grid//.//$year $hour_mean  # statistics file
998                          # error value
$IDWweight      # weighting of the reciprocal distance for IDW
0.75              # for method 3: relative weight of IDW
$IDWmaxdist     # max. distance of stations
$Anisoslope     #  slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie    #  ratio of short to long axis of the anisotropy-ellipsis
0.1                  # lower limit of interpolation results
0                    # replace value for results below the lower limit
900                  # upper limit for interpolation results
900                  # replace value
$SzenUse    # 1=use scenario data for correction, 0=dont use scenarios
3       # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1       # number of scenario cells
699000 235000    0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

2.3.11 linear combination of IDW and EDRINT (method 11)

This method was already described in detail in chapter 2.3.3 . This is a reference to this section only.

Practical application:

Control file: (red, bold and large font: specific settings for method IDW)

```
[precipitation]
11                              # method
$inpath//regen//$year//.dat AdditionalColumns=0   # station data
820 1400 200 1 300 # lower inversion [m asl], upper inversion
#   [m   asl],   tolerance   [m],   overlap   [0/1   for   true/false],
#clusterlimit [m]
$outpath//idw_//$preci_grid  # name of the output grid
$Writegrid                   # writecode
0.1                          # correction faktor for results
$outpath//idw_prec//$grid//.//$year $hour_mean  # statistics file
998                # error value
$IDWweight      # weighting of the reciprocal distance for IDW
0.75              # for method 3: relative weight of IDW
$IDWmaxdist     # max. distance of stations
```

```
$Anisoslope      #  slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie     # ratio short to long axis of anisotropy-ellipsis
0.1              # lower limit of interpolation results
0                # replace value for results below the lower limit
900              # upper limit for interpolation results
900              # replace value
$SzenUse    # 1=use scenario data for correction, 0=dont use scenarios
3       # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1       # number of scenario cells
699000 235000    0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

2.3.12 Nearest neighbor combined with a constant lapse rate (method 12)

This method is particularly useful for sparse stations when elevation dependent regressions for temperature data is not reliable enough. Then, a fix lapse rate may be used. Using the stations elevation as base, the temperature (or any other entity) may be interpolated using the elevation difference and the lapse rate.

$$T(h) = T_{\text{nearest_station}} + r(h - h_{\text{neatres_station}}) \qquad \textbf{2.3.13}$$

with $T(h)$ result of the linear combination of station value and applied lapse rate

$T_{\text{nearest_station}}$ is the entities value at the station nearest to the actual location

r is the lapse rate (unit per meter elevation, e.g. K/m)

h is the local elevation of the actual cell in m.a.s.l.

$h_{\text{nearest_station}}$ is the elevation of the nearest station in m.a.s.l.

There are two variations possible: The basic variant is the simple lapse rate, valid over the entire input data and elevation range. Then there is the variation of using two different lapse rates for two elevation bands. If this variation is used, another variation can and must be used as well: the data type must be defined: Is the data rather a continuous type like temperature or is it rather a discontinuous type like precipitation.

When using one of these variations, the specific parameters are extended. Otherwise (basic version), there is only one specific parameter: the lapse rate in <Delta of measurement of entity>/m, e.g. K/m for temperature or mm/m for precipitation. The extended variations use three additional parameters:

- the threshold elevation $h_{threshold}$ for switching from the first to the second lapse rate [m]
- the second lapse r_{upper} [change of entity per unit elevation change, e.g. mm/m for precipitation]
- the data type: P-type for discontinuous entities (e.g. precipitation) and T-type for continuous entities (e.g. temperature). For discontinuous entities, there will be no change by lapse rate when the input data was 0 or less (otherwise a precipitation of 0 mm would cause positive precipitations for positive lapse rates and even negative precipitations for negative lapse rates).

When the threshold elevation for swithcing between the lapse rates is reached, the entity above that threshold elevation is calculated by

$$T(h > h_{threshold}) = T_{\text{nearest station}} + r_{lower} \cdot (h_{threshold} - h_{\text{nearest station}}) + r_{upper}(h - h_{\text{threshold}}) \qquad (2.3.14)$$

Practical application:

Control file: (red, bold and large font: specific settings for method IDW)

```
[temperature]
12                              # method
$inpath//temper//$year//.dat AdditionalColumns=0   # station data
0.007 2500 0.01 T-type # method 12, here the temperature lapse
#rate is higher above 2500 m above sea level than below
#alternative (simple or basic version)
#0.007 # method 12 for one lapse rate only
$inpath//temper//$year//.out  # regression data
$outpath//idw_//$preci_grid  # name of the output grid
$Writegrid                   # writecode
0.1                          # correction faktor for results
$outpath//idw_prec//$grid//.//$year $hour_mean  # statistics file
998                 # error value
$IDWweight          # weighting of the reciprocal distance for IDW
0.75                # for method 3: relative weight of IDW
$IDWmaxdist         # max. distance of stations
$Anisoslope         #  slope of the anisotropy-ellipsis (-90+ to +90°)
$Anisotropie        # ratio short to long axis of anisotropy-ellipsis
0.1          # lower limit of interpolation results
0                   # replace value for results below the lower limit
900                 # upper limit for interpolation results
900                 # replace value
$SzenUse    # 1=use scenario data for correction, 0=dont use scenarios
3       # 1=add scenarios, 2=multiply scenarios, 3=percentual change
1       # number of scenario cells
699000 235000    0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
# coordinates of the cells, then one value for each month of a year
```

2.3.13 Using multiple interpolation methods at the same time (Regional Superposition)

The definition of interpolators is generally not limited by WaSiM. It is possible to define as many interpolation sections (each with its own identification string) as the hardware will be able to process. However, the internal identification strings are limited to the following list:

- global_radiation used in evaporation and snow/glacier melt
- net_radiation used in evaporation and snow/glacier melt
- sunshine_duration used in evaporation
- temperature used in evaporation, snow/glacier melt, permafrost thawing and dynamic phenology
- temperature_14 used for evaporation (Wendling)
- wind_speed used for evaporation
- vapor_pressure used for various evaporation approaches
- vapor_pressure_14 used for evaporation (Wendling)
- air_humidity used for various evaporation approaches
- air_humidity_14 used for evaporation (Wendling)
- air_pressure not used yet
- precipitation input for interception/snowmelt/soil model etc.

Multiple interpolation methods for the same data type can be used in different ways:

1. Using two (or more) methods for the entire basin and superposing the results with specific weights for each method. Prior to Version 8.5 this was already possible with interpolation method 3 (and now also 11), but for IDW and EDR(EXT) only.
2. Using specific interpolation methods for different regions with a smooth transition. This was

partly possible prior to version 8.5: Both methods must be EDR, but there could be as many as up to 30 regions using specific parameter files each. Starting with version 8.5, any of the interpolation methods described above may be used for any of the regions defined in the region grid

3. It is even possible to combine both uses: multiple interpolation methods can be used for each region with different weights, offering the greatest flexibility thinkable at this time.

A special feature is the smooth transition of the interpolation results between regions.

Defining Regions:

The regions are given as a grid with the same geometry as the zone grid (same numbers of rows and columns, same cell size, nodata value, lower left x- and y-coordinates). Each region is identified by a unique code. The codes should be integer numbers of any order, no ranking is required, gaps are allowed (e.g. if using codes 4, 1 and 8 following any special notation).

The regions are technically totally independent on the zones or subbasins, except for the fact, that each valid grid cell of the zone grid must belong to one of the regions. Zones are used for the statistic output of the model results whereas regions are used for selecting the matching interpolator. Near the borders between the regions the model produces a smooth transition by weighing the regression interpolation results of up to three regions dependent on the distances of the actual cell to the respective regions.

If there are multiple regions defined, the model performs the following tasks:

Tasks at initialization:

- Reading the region grid (the identification string for the region grid in the control file has to be “regression_regions“ in lower cases without quotes).
- Reading the transition distance, i.e. the maximum distance of a cell to any regions border to be affected also by the neighboring region(s). This parameter is given in meters in the new section [region_transition_distance] of the control file.
- Generating a REGION2- and a REGION3-grid. The REGION2 grid contains for each grid cell the region code of the second region the cell is affected by. If the grid cell is not within the maximum transition distance from any region border, than the REGION2 grid contains a zero-value in the matching location. In Analogy to REGION2, the REGION3 grid contains the codes of the third region the cell is affected by, also only if there is a third affecting region at all. The REGION2 and REGION3 grids are written into the directory the region grid was read from with the file extensions „.$r2“ and „.$r3“, respectively, for the modellers reference.
- Generating a WEIGHT1, WEIGHT2 and WEIGHT3-grid containing the weights the respective regions have for the interpolation result. When doing the interpolation, for each grid cell the up to three region grids are checked. If the matching positions contain codes greater than zero, the actual result is scaled with the weight of the according WEIGHT<n> grid. The weights are calculated after the following scheme. If a cell in row *i* and column *j* is affected by only one region (coded in the grid REGION1), the weights are $WEIGHT1_{i,j} = 1.0$, $WEIGHT2_{i,j} = 0$ and $WEIGHT3_{i,j} = 0$ for the regions indexed in the matching grids $REGION1_{i,j}$, $REGION2_{i,j}$, and $REGION3_{i,j}$, respectively.

  region weights for cells affected by two regions

  Weights for cells affected by two regions are calculated using a linear transition starting with a weight of 1.0 if the cell is at the maximum transition distance from the region border and decreasing to 0.5 if the cell is directly at the border. This weight is stored in $WEIGHT1_{i,j}$, the weight of the neighboring regions regression result is then simply $WEIGHT2_{i,j} = 1.0-$

WEIGHT1$_{i,j}$.

$$w1_{i,j} = 0.5 + 0.5 \cdot d / r$$
$$w2_{i,j} = 1.0 - w1_{i,j} \qquad \textit{(if only 2 regions affect the actual cell)} \qquad \textbf{(2.3.14)}$$

with $w1_{i,j}$ weight of the regression result using the parameters for the region the cell belongs to

$w2_{i,j}$ weight of the regression result using the parameters for the second region the actual cell is affected by

d distance of the actual cell to the region border [m]

r maximum distance to be considered for smooth transitions, specified in the control file in the section [region_transition_distance], unit [m]

region weights for cells affected by three or more regions

If a cell is affected by three ore more regions, only the three nearest regions are used for calculating weights. The sketch in figure 2.3.13.1 illustrates the calculation of weights. The maximum distance from any region border a point must have to be not influenced by another region is the radius *r*.

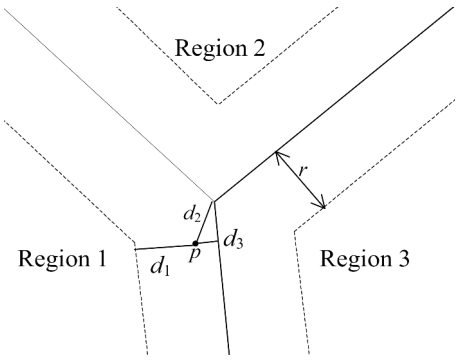


figure 2.3.13.1: Weighting of multiple regions for a point p near the regions crossing point

If the interpolation point is only affected by one neighboring region, the above formula (2.3.14) is applied. Only if the crossing point of three regions is within the distance *r* around the interpolation point, two neighboring regions are considered for calculating the weights. The weights are calculated by a modified inverse distance weighting scheme. The following constraints must hold:

- (1) Regardless of the shape of the region borders, a point at the crossing point of three regions have to have a weight of 1/3.
- (2) The weight of a region decreases with increasing distance *d* of the point to that region until it reaches the value of 0.0 at the distance $d = r$.
- (3) The rate of increase of the weights should be linearly related to the decreasing distance to the nearest region in order to match the linear transition of weights between two regions at the outer limit of the three-region-affected area.
- (4) The sum of all three weights must be 1.0.

In a first step the distances d_2 and d_3 are extended by *r* in order to satisfy condition (1) resulting in:

$$u_1 = d_1$$
$$u_2 = r + d_2 \qquad \textbf{(2.3.15)}$$
$$u_3 = r + d_3$$

If $d_2 = d_3 = 0$, then $u_1 = u_2 = u_3 = r$ resulting in equal weights of 1/3 for all regions. To satisfy conditions (2) and (3), the inverted distances u_1 to u_3 are modified in the following way:

$$x_1 = \frac{1}{u_1} - c \cdot \frac{1}{2r}$$
$$x_2 = \frac{1}{u_2} - c \cdot \frac{1}{2r} \qquad \textbf{(2.3.16)}$$
$$x_3 = \frac{1}{u_3} - c \cdot \frac{1}{2r}$$

with $x_1 ... x_3$ modified inverse distances from the interpolation point to the three regions of interest

$u_1 ... u_3$ distances d_1 to d_3 to the three regions after figure Fehler: Verweis nicht gefunden, extended by r after equation (2.3.15).

r maximum distance of the interpolation point to another region to be affected by this region

c correction factor for modifying the inverse distances

The correction factor c depends on the minimum distance of the interpolation point to any regions interior area (usually of region 1) and is responsible for satisfying condition (3). It is estimated after:

$$c = 0.9 - \frac{1}{2} \cdot \cos\left(\frac{\pi}{2} \cdot \frac{\min(u_1, u_2, u_3)}{r}\right) \qquad \textbf{(2.3.17)}$$

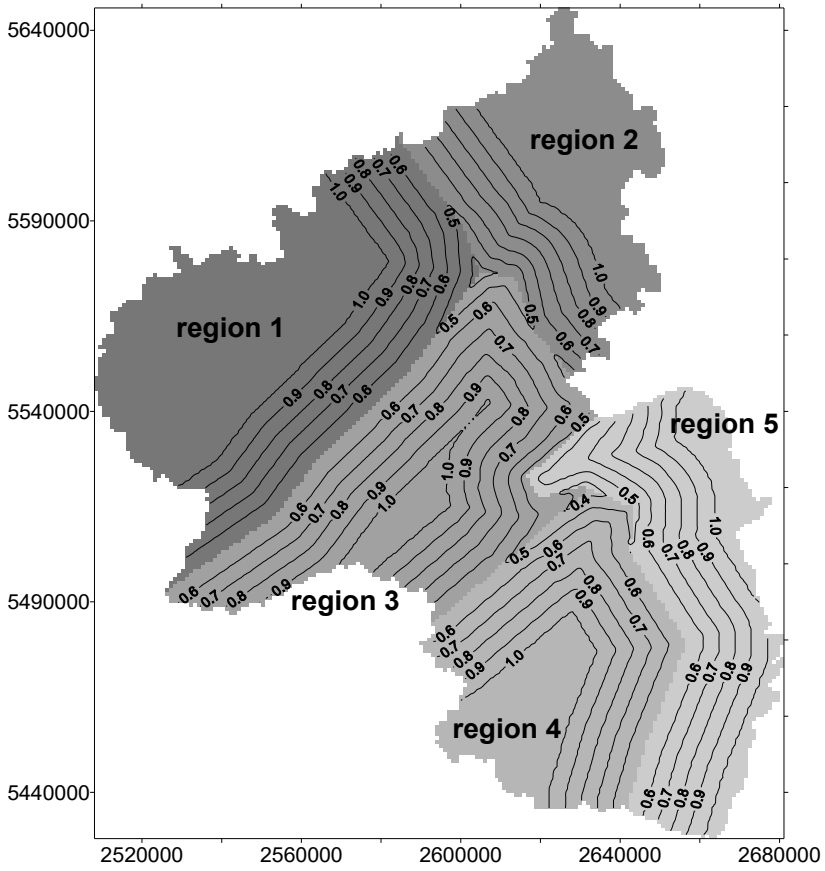


figure 2.3.13.2Weighting of the actual region if using multiple regions. Rheinland-Pfalz, Germany, 5 regions, maximum transition distance r was set to 20 km.

The weights of the regions are then calculated by the usual inverse distance weighting scheme:

$$w_i = \frac{1}{d_i} \cdot \frac{1}{\sum_{i=1}^{3} d_i} \qquad (i=1..3) \qquad \textbf{(2.3.18)}$$

with w_i weighting of the regions 1...3 for the modified inverse distance interpolation

Figure 2.3.13.2 shows the weights for the regression parameter sets of the regions 1 to 5 for cells which belong to the respective regions. Such weighting grids are also generated for the second and third region affecting the actual regions. The sum of all three weighting grids is 1.0 for all valid region cells.

Tasks at interpolation:

- Each time the interpolation is done, the region grids $REGION1_{i,j}$, $REGION2_{i,j}$, and $REGION3_{i,j}$ are checked for entries unequal to zero for each location *i,j*. Region grid REGION1 (the only grid the model reads in at initialization time) must contain valid entries for each grid cell which is valid in the zone grid, whereas the other region grids (which are internally created during initialization) contain only valid entries, if the cell is affected by other regions when its location is within the maximum transition distance between two or more regions *r*.
- Calculating the interpolation results by weighing the results of the at maximum 3 affecting regions. **Note: each single result may already be the result of a superposition of multiple interpolation methods. Here, only the generation of a smooth transition between regions is described. The superposition itself is described later in this chapter.**

- Combining the results by using the region weights for the actual location:

$$z = w_1 z_1 + w_2 z_2 + w_3 z_3 \qquad \textbf{(2.3.19)}$$

with z weighted result of altitude dependent regression for the actual location
$z_1 ... z_3$ results of the altitude dependent regressions for the three regions affecting the actual location
$w_1 ... w_3$ weights for the three regions after equation (2.3.18)

Defining rules for weighted superposition:

- The region grid must be read in as a standard grid, it may have any file name, but the identification string must be *regression_regions*, the fill parameter must be 1:

```
[standard_grids]
...
$inpath//$grid.reg      regression_regions 1     # regression regions
...
```

- The section [region_transition_distance] must be created in the control file containing the maximum distance, a cell can be located from any other region to be affected by this other region:

```
[region_transition_distance]
10000       # maximum transition distance in meter
```

- The control file must contain a new section [RegionalSuperposition]. An example is given below. This section contains the main switch (0 or 1), the time step, the numbers of superpositions to be carried out and then for each of the superpositions a description starting with the resulting entity name (e.g. precipitation). Each input from an interpolator must be identified by the name given to this interpolator (e.g. precipitation_reg) and a relative weight must be defined for each region (when no regions are used, then default region 1 must be attached a weight of 1.0). After the list of entityinputgrids the name of the outputgrid and the statistics file with its common writecodes must be given.

```
[RegionalSuperposition]
1
$time
NumberOfEntities = 2;
precipitation {
      entityinputgrid = precipitation_reg ;
            regions = 1   2   ;
            weights = 0.0 0.2 ;
      entityinputgrid = precipitation_idw ;
            regions = 1   2   ;
            weights = 1.0 0.8 ;
      outputgrid  = $outpath//$preci_grid ;
            writecode = $Writegrid ;
      outputtable = $outpath//prec//$grid//.//$year;
            statcode  = $hour_mean;
}
temperature {
      entityinputgrid = temperature_reg;
            regions = 1   2   ;
            weights = 0.7 0.8 ;
      entityinputgrid = temperature_idw;
            regions = 1   2   ;
            weights = 0.3 0.2 ;
      outputgrid  = $outpath//$tempegrid ;
            writecode = $Writegrid ;
      outputtable = $outpath//temp//$grid//.//$year ;
            statcode  = $hour_mean ;
}
```

Note 1: all weights must add up to 1.0 for each single region

Note 2: all regions must be listed in the regions list for each input grid. If a specific interpolation should not be considered for a region, then the weight must be set to 0.

Note 3: there may be as many interpolators combined as required. It's not limited to 2 methods only. Even the same method with different input data could be used (this is the equivalent to the old method for regression only).

Note 4: When using multiple regions with superpositions, precipitation correction will be carried out AFTER interpolation and superposition of interpolation results because it is not predictable before superposition runs what values temperature and wind speed will have at the stations location. Be aware of artifacts (steps) due to non-linearities in the precipitation results in case of temperatures in the range around the threshold for snow/rain. Since another correction for snow may apply, the resulting maps may show steps where the temperature suddenly falls below the snow/rain threshold (which was not the case when correcting the precipitation at station locations only). It's recommended to apply precipitation correction prior to interpolation, i.e. to feed WaSiM with already corrected precipitation.

Note 5 : the separate interpolators should be named in a speaking manner, like precipitation_idw for an interpolator with method IDW and precipitation_reg for an interpolator with method regression. The definition of the superpositions will then define the name as expected in the model (see list with allowed names). Precipitation must then always be named as precipitation, temperature as temperature or temperature_14 etc. (the submodels expect the precipitation, temperature etc. to have exactly these names).

Examples:

Figure 2.3.13.3 shows the region grid for the river Thur basin (Switzerland). The northern part (in blue) is the lower region, the southern part (in red) it the mountainous region. Here, the regions are defined following sub basin borders (cell size: 500m). However, this is not required. Regions may have any shape.

Figures 2.3.13.4 to 2.3.13.7 show the superposition results for temperature with varying transition zones: 0km (no transition) for figure 2.3.13.4, 1km for figure 2.3.13.5, 5km for figure 2.3.13.6 and finally 10km for figure 2.3.13.7. The IDW and EDR interpolations were done for the entire basin. During the regional superposition, the local values of each interpolation method were weighted according to the description:

```
 temperature {
      entityinputgrid = temperature_reg;
            regions = 1   2   ;
            weights = 0.0 1.0 ;
      entityinputgrid = temperature_idw;
            regions = 1   2   ;
            weights = 1.0 0.0 ;
      outputgrid  = $outpath//$tempegrid ;
            writecode = $Writegrid ;
      outputtable = $outpath//temp//$grid//.//$year ;
            statcode  = $hour_mean ;
}
```

This means, that for the northern part Only IDW is used whereas for the southern part only EDR is used. The transition zone, however, uses the results of both methods according to the weights of the affected regions.

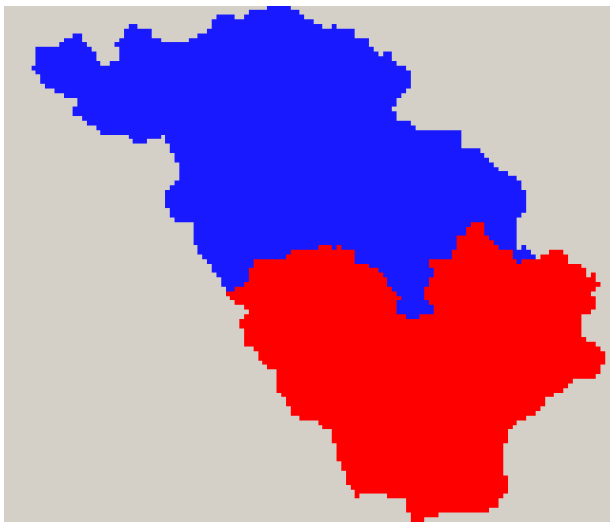


figure 2.3.13.3: Definition of two regions for the river Thur basin (Switzerland, 1700km2), the entire grid encompasses an area of 65.5 km x 56 km, each cell is 500 by 500m (also for all following figures)

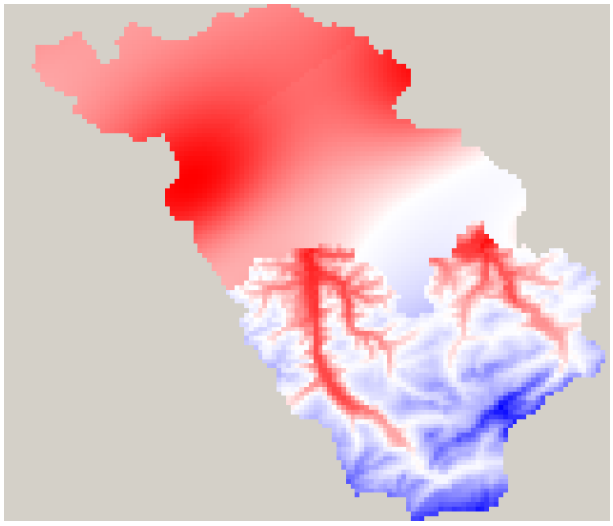


figure 2.3.13.4: interpolation results for IDW (North) and EDR (South) without smooth transition

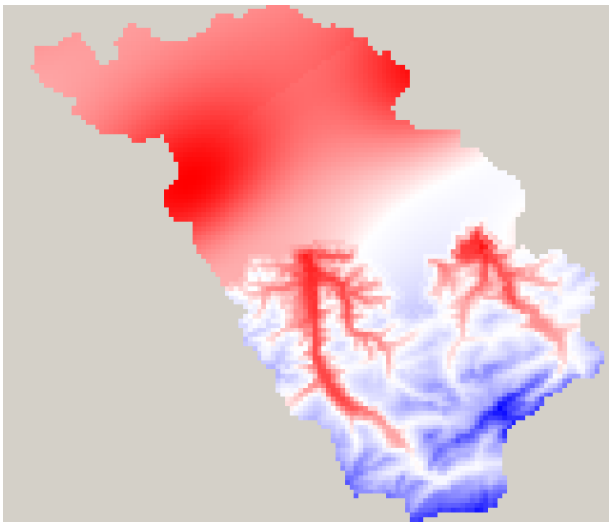


figure 2.3.13.5: interpolation results for IDW (North) and EDR (South) with 1km transition-range

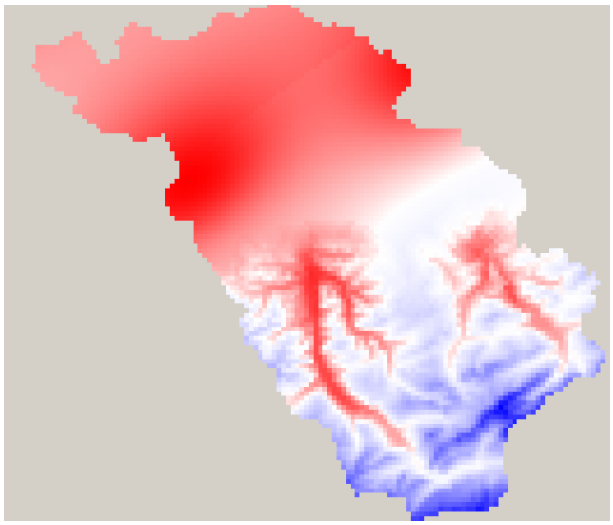


figure 2.3.13.6: interpolation results for IDW (North) and EDR (South) with 5km transition-range

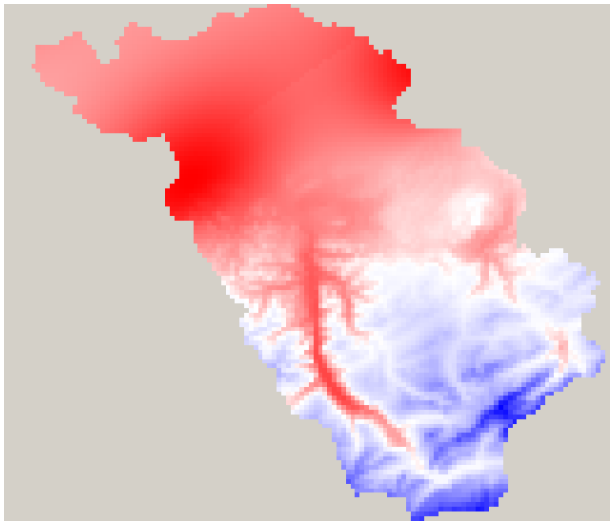


figure 2.3.13.7: interpolation results for IDW (North) and EDR (South) with 10km transition-range

2.3.14 Tolerating additional columns in meteorological input files

Additional columns in the meteorological input files may appear after the four columns for the date (YYYY, MM, DD, HH). The reasons for the existence of such columns may be, that the input file is generated as an output of a GCM. Such additional columns must be specified in the following way. In the control file each input file is specified with ist name. If a clause like "AdditionalColumns=2" follows this file name, then two additional columns are expected. There are no spaces ar tabs allowed before or after the equal-sign.

Example:

```
[temperature]
2                              # Methode 1=idw, 2=regress, 3=idw+regress, 4=Thiessen
$inpath//tempe//$year//a.dat AdditionalColumns=2 # station data and # of add. columns
$inpath//temper//$year//.out  # file name with regression data (if method = 2 or 3)
```

The given number of additional columns will be neglected in each row of the input file. They

MUST appear in each row after the 4 date-columns.

2.3.15 Optimized Smooth Interpolations for Larger Areas With Multiple Regions

When running multiple WaSiM models in neighbouring regions and displaying interpolation results or other grid content in one single picture, the different interpolation settings that were optimized for each single basin will usually lead to sudden changes at the borders of the basins. Such a situation is clearly visible in figure 2.3.15.1.

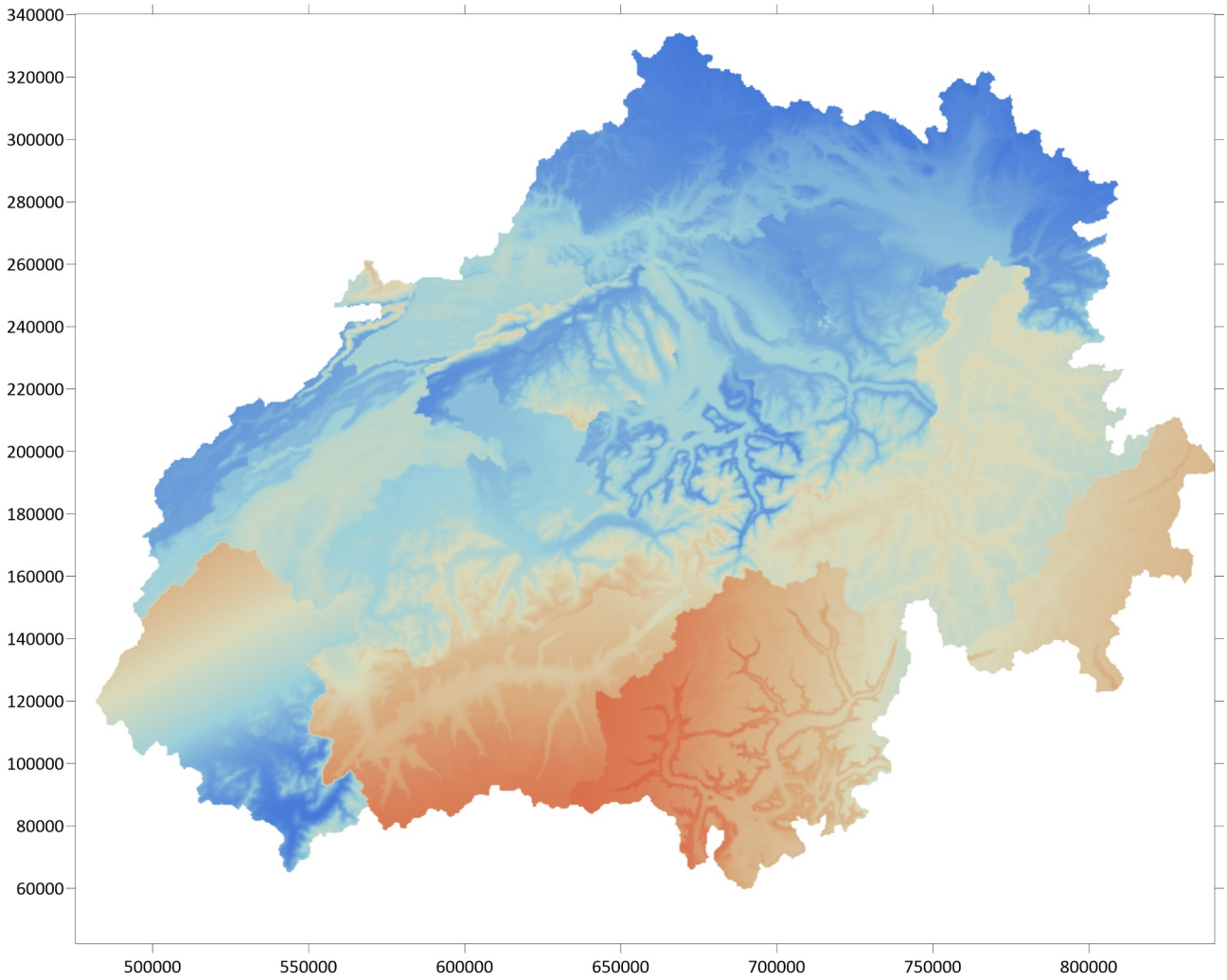


figure 2.3.15.1: Example of relative air humidity interpolation results for all Swiss river basins, interpolated independently of each other

The picture is schematic only, so no absolute values are shown in the legend. In fact, it is the average relative air humidity for 1991 (entire year).

The much better picture is shown in figure 2.3.15.2. Although this is only the average value for January 1991, the difference is clearly visible: No sudden changes but smooth transitions between the basins result in a much better overall picture.

This effect is not new to WaSiM. By using regions and a definition of a suitable distance in section [region_transition_distance], all interpolation results are overlayed using the distance from the basin border as an additional parameter.

However, when running the interpolation for a very large area with many regions, the overall model performance is important. The classic method runs each interpolation for the entire model area. Here, the new model version allows to define a region for each interpolator, so that in the end the model runs much faster.

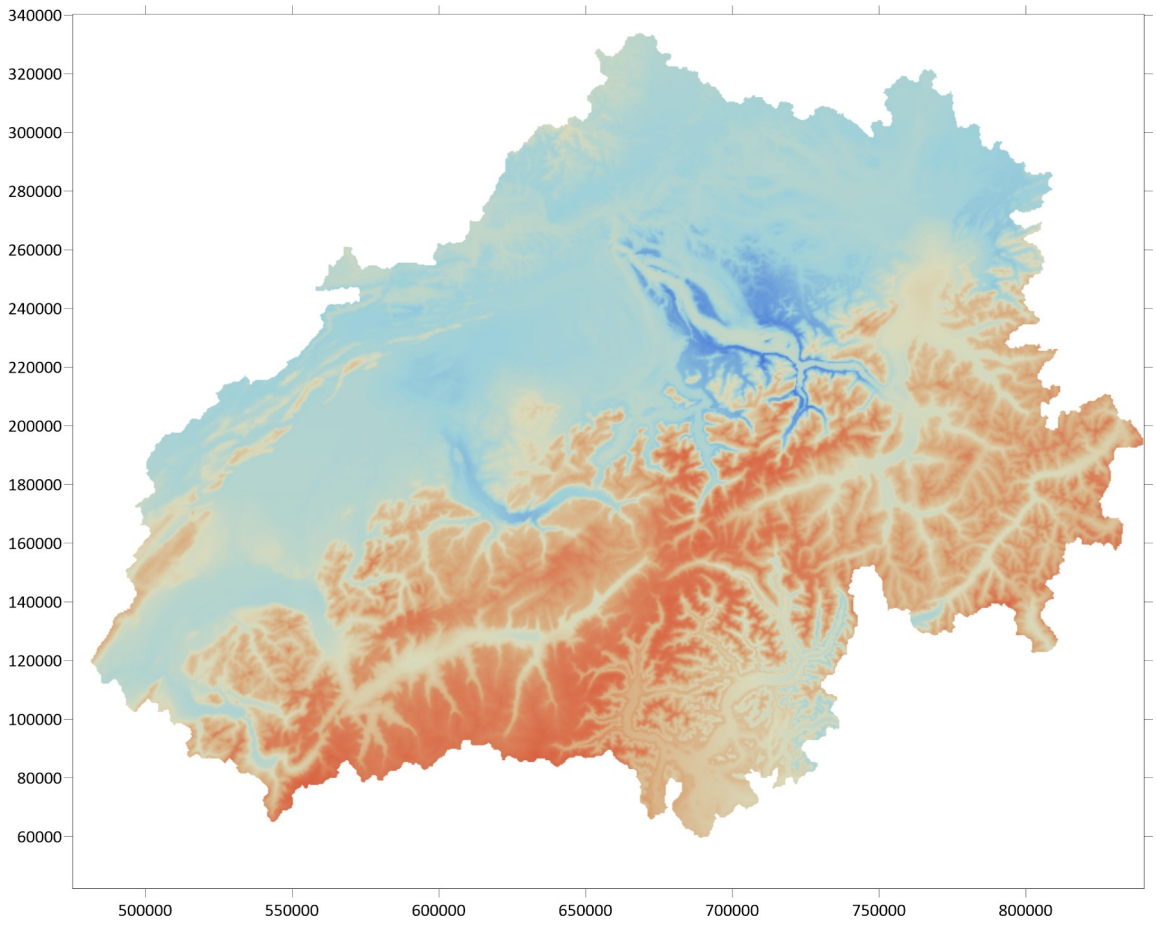


figure 2.3.15.2: Example of relative air humidity interpolation results for all Swiss river basins, interpolated independently of each other but with smooth shifts between regions

The main new feature is activated by adding the coordinates of the window of interest in the control file like this:

```
[temperature_region1]
10                                                  # methods, see comments above
$inpath_meteo//Region01/T_1991-2020.dat AdditionalColumns=0 xmin=670000 ↵
xmax=840500 ymin=122000 ymax=282500 # input file with window coordinates
820 1400 200 1 300 # lower inversion [m asl], upper inversion [m asl]
$outpath//$tempegrid1          # name of the output grid
...
```

Some notes on interpolations and regions:

1. WaSiM can read in grids with already interpolated values. The read in grid should cover the entire region of the interpolator, otherwise an IDW interpolation will be used for selecting the actual value.
2. If for method 9 and others (read in grids) the grid dimensions and coverage differs from the WaSiM internal model grid, WaSiM will interpolate by bi-linear interpolation.
3. If the read in grids have the same dimensions and coverage as the internal model grids, then this bi-linear interpolation is equivalent with taking the actual value directly

4. When using multiple regions, the interpolators need to use specific names for their results that must not be identical to the internally used names (i.e. the names temperature, precipitation, global_radiation, sunshine_duration, air_humidity, and wind_speed mustn't be used for the interpolators
5. After interpolating different regions with or without the new "windows of interest"-coordinates, the results must be combined in section [superposition] with appropriate weights. Here, the resulting entities must have the names that WaSiM uses internally (i.e. temperature, precipitation, global_radiation, sunshine_duration, air_humidity, and wind_speed).

The above examples were interpolated by using 22 regions for precipitation, 18 for temperature and 17 for all the other entities. In total, there were 108 interpolators defined, so the control file describing the interpolations and superpositions contains more than 3000 lines.

The performance boost by using the windows of interest was around 100% (i.e. twice as fast as without such a limiting window).

2.4 Applying scenarios to the meteorologic driving data

2.4.1 Applying scenarios with low spatial resolutions

As any other hydrologic model, WaSiM is often used to simulate hydrologic responses to climate change. The main problem herewith is, that no future climate data is available in the required spatial and temporal resolution and that, even when they were available, the model results wouldn't be comparable with actually observed runoff data. Often, GCM-results are stored in low spatial resolution and even lower temporal resolution (However, with more advances in computer technology and storage capacity, it becomes more and more common to use GCM-outputs directly as input to hydrologic models or to use re-analysis data of GCM experiments and down scaling techniques to get input data series which are at least statistically comparable to actual statistics).

One very common method to apply climate change scenarios is to run the model with historical data and to apply the changes in the GCM scenario runs compared to the control runs. These changes can be added (e.g. temperature) or multiplied (e.g. precipitation) to the actually observed meteorologic input data.

To enable WaSiM to use scenarios, each meteorologic entity has a switch which must be turned on (1) to activate scenario use (see red lines in the example below). The next line contains the method, how the scenarios should be applied to the interpolated data:

- method 1: add the interpolation data
- method 2: multiply the scenario data
- method 3: apply scenario data as percentage change (which is in fact like doing a multiplication)
- method 5 and 6: applying scenario grids, see next chapter

To account for spatial and temporal differences in the scenarios, any number of scenario locations may be defined (which is the next parameter → number of scenario cells). Each following row contains then the coordinates (x and y coordinates corresponding to the model domain) and 12 scenario values for the 12 months of a year.

```
[vapor_pressure]
10  # method; 10 = internal elevation dependent regression
$inpath//dampf//$year//a.dat AdditionalColumns=0  # file name with station data
820 1400 200 1 300 # regression parameters
$outpath//$vaporgrid          # name of the output grid
$Writegrid                    # write code for the output grid
0.1                           # correction faktor for results
$outpath//vapo//$grid//.//$year $hour_mean # file name for the statistic output
998                           # nodata indicator (values <-998 and >998 are interpreted as nodata)
$IDWweight                    # weighting of the reciprocal distance for IDW
0.5                           # for method 3: relative weight of IDW-interpolation in the result
$IDWmaxdist                   # max. distance of stations to the actual interpolation cell
$Anisoslope                   #  slope of the mean axis of the anisotropy-ellipsis (-90 ... +90)
$Anisotropie                  #  ratio of the short to the long axis of the anisotropy-ellipsis
0                             # lower limit of interpolation results
0                             # replace value for results below the lower limit
90                            # upper limit for interpolation results
90                            # replace value for results with larger values than the upper limit
1                             # 1=use scenario data for correction, 0=dont use scenarios
1                             # 1=add, 2=multiply, 3=% change, 5/6=monthly scaling grids 5=+, 6=*
4                             # number of scenario cells
699000 235000  0.994 1.187 1.021 1.035 1.201 1.021 .635 .566 .538 1.021 .800 1.007
699000 290000  1.021 1.201 1.035 1.049 1.201 1.035 .649 .566 .538 1.035 .787 1.021
737000 235000  0.980 1.173 1.035 1.076 1.173 1.021 .621 .635 .593 1.035 .800 1.007
737000 290000  1.007 1.187 1.049 1.090 1.173 1.035 .649 .649 .607 1.049 .800 1.007
```

2.4.2 Applying monthly scenario grids

When WaSiM is driven by results of GCM´s which are interpolated using bi-linear interpolation it may be advantageous to consider known small scale variability by scaling the interpolation results by a spatially varying factor. Thus, the known variability of e.g. monthly rainfall, or sunshine duration, or radiation etc. can be lay over the rather smooth interpolation results. Since there was already a possibility in WaSiM to scale the interpolation results by monthly scenarios for given locations, and since the consideration of scaling grids can be seen as a more detailed way of applying scenarios, the declaration of scenario grids is consequently done in the control file sections for interpolating meteorological data:

```
[temperature]
…
$IDWweight    # weighting of the reciprocal distance for IDW
0.1             # for method 3: relative weight of IDW-on the result
$IDWmaxdist   # max. distance of stations to the actual interpolation cell
$Anisoslope   # slope of main axis of the anisotropy-ellipse…
$Anisotropie  # axis ratio of the anisotropy-ellipse
-40                   # lower limit of interpolation results
-40                   # replace value for results below the lower limit
40                    # upper limit for interpolation results
40                    # replace value for results above the upper limit
1                     # 1=use scenario data, 0=dont use scenarios
5          # 1=add, 2=multiply, 3=percentual change, 5=add monthly grids, 6=multiply monthly grids
$inpath//scalegrd.01    # scaling grid for small scale variability month 1
$inpath//scalegrd.02    # scaling grid for small scale variability month 2
$inpath//scalegrd.03    # scaling grid for small scale variability month 3
$inpath//scalegrd.04    # scaling grid for small scale variability month 4
$inpath//scalegrd.05    # scaling grid for small scale variability month 5
$inpath//scalegrd.06    # scaling grid for small scale variability month 6
$inpath//scalegrd.07    # scaling grid for small scale variability month 7
$inpath//scalegrd.08    # scaling grid for small scale variability month 8
$inpath//scalegrd.09    # scaling grid for small scale variability month 9
$inpath//scalegrd.10    # scaling grid for small scale variability month 10
$inpath//scalegrd.11    # scaling grid for small scale variability month 11
$inpath//scalegrd.12    # scaling grid for small scale variability month 12
0                       # number of scenario cells
```

For each month of the year, an offset grid (scenario method 5) or scaling grid (scenario method 6) has to be specified. The grids are sequentially read in during the model run each time when a new month starts. There is no temporal interpolation between different grids respectively between different months, so abrupt changes in the scenario grid will cause also abrupt changes in the temporal course of the scaled interpolation results.

2.5 Topography dependent adjustment of radiation and temperature

2.5.1 Calculation of sun coordinates, sunset and sunrise, radiation correction

The impacts of the topography on radiation and air temperature is considered using the scheme after OKE (1987). The symbols meaning are:

- ϕ geographical latitude (positive north, negative south)
- δ declination of the sun (angle between sun and the equatorial plane of the earth)
- Z zenith angle (angle between the direction to the sun and the local zenith)
- h_s hour angle (angle, the earth must rotate to bring the meridian directly below the sun)
- Ω_s azimuth angle (angle between the projection of the direction to the sun and the north direction)
- t_J Julian day (1st of January = 1, 31st of December = 365)
- t true local sun time (from average sun time corrected using: Jan: -3.2, Feb: -13.6, Mar: -2.6, Apr: -4.2, May: +2.8, Jun: +2.5, Jul: -3.5, Aug: -6.3, Sep: -0.3, Oct: +10.0, Nov: +16.4, Dec: +11.3 min)

Θ_s incidence angle between the normal to the local slope and the direction to the sun
β_t local slope angle
$\hat{\beta}$ height of the sun above the horizon (as angle)
$\hat{\Omega}$ azimuth angle of the local slope (aspect, clockwise from north=0 to north=360)

The equations for calculating the zenith and azimuth angles are given by:

$$\cos Z = \sin\phi \sin\delta + \cos\phi \cos\delta \cos h_s = \sin\beta_t \qquad \textbf{(2.5.1)}$$

$$\cos\Omega_s = (\sin\delta \cos\phi - \cos\delta \sin\phi \cos h)/\sin Z \qquad \textbf{(2.5.2)}$$

$$d = \text{-}23.4 \cdot \cos[360(t_J + 10)/365] \qquad \textbf{(2.5.3)}$$

$$h_s = 15(12 - t) \qquad \textit{in Grad} \qquad \textbf{(2.5.4)}$$

with
$$\Omega_s = \arccos(\cos\Omega) \qquad (t < 12,\ h_s > 0)$$
$$\Omega_s = 360° \text{-} \arccos(\cos\Omega) \qquad (t > 12,\ h_s < 0)$$
$$\Omega_s = 180° \qquad (t = 12,\ h_s = 0)$$

The relations for calculating sun rise and sun set for use within the shadowing routine and within the evaporation routine are given by:

$$\cos h_s = \frac{\sin\phi \sin\delta}{\cos\phi \cos\delta} \quad \rightarrow \quad h_{sr} = \frac{12}{p} \cdot \arccos(\cos h_s) = \frac{12}{p} \cdot h_s \quad \text{and} \quad h_{ss} = 24 - h_{sr} \qquad \textbf{(2.5.5)}$$

with h_{ss} time of sun set [h]
h_{sr} time of sun rise [h]

The incoming clear sky direct shortwave radiation I_{max} on a plain perpendicular to the direction to the sun can be estimated using an approach for the atmospheric extinction given by OKE (1987):

$$I_{max} = I_{tof} \cdot \psi_a^{\frac{p/p_0}{\cos Z}} \qquad \textit{in } W \cdot m^{-2} \qquad \textbf{(2.5.6)}$$

with I_{tof} incoming shortwave radiation at the top of atmosphere [W·m^{-2}]
(solar constant ≈ 1368 W·m^{-2})
p air pressure for the place of interest [hPa]
p_0 air pressure at sea level (≈ 1013 hPa)
ψ_a transmissivity of the atmosphere [-]

A horizontal plane will then get a normal irradiation I_{norm} given by:

$$I_{norm} = I_{max} \cdot \cos Z \qquad \textbf{(2.5.7)}$$

This type of irradiation estimation is used also in the evaporation module if no sunshine duration data is available.

The angle between the direction to the sun and the normal to a given grid cell (the incidence angle $\hat{\Theta}$) and hence the correction factor for the direct irradiation is given by:

$$\cos\hat{\Theta} = \cos\beta_t \cdot \cos Z + \sin\beta_t \cdot \sin Z \cdot \cos(\Omega_s - \hat{\Omega}) \qquad \textbf{(2.5.8)}$$

For an horizontal plane the correction is then given by:

$$I_{korr} = I_{norm} \cdot \frac{\cos\hat{\Theta}}{\cos Z} \qquad \textbf{(2.5.9)}$$

This correction factor can be applied only with some restrictions:

1. the sun have to be risen over the horizon,
2. the elevation angle of the sun should be greater than 12°, because smaller angles results in a to fast decreasing optical density of the atmosphere (exponent in equation (2.5.9)); if the sun is less than 12° above the horizon, the correction factor is limited to ± 5.
3. the considered grid cell may not be in the shadow of another cell and $\quad < 90°$ must hold.
4. the relative sunshine duration has to exceed 0.1, otherwise only diffuse radiation is assumed which will not be corrected

If the model time step is to large for a reasonable radiation correction (e.g. if using daily time steps) the time step is subdivided into sub-time-steps. Then, a radiation correction is carried out for each of these sub-time-steps. The effective correction factor for the entire time step is then found as the weighted average of all sub-time-step correction factors. The weights for each sub-time-step are then calculated by the fraction of the normal irradiation for the sub-time-step of the normal irradiation for the entire time step which is the sum of the normal irradiation of all sub-time-steps.

$$f_d = \frac{\sum_{i=1}^{n} \frac{\cos\hat{\Theta}_i}{\cos Z_i} I_{norm,i}}{\sum_{i=1}^{n} I_{norm,i}} \qquad \textbf{(2.5.10)}$$

with f_d average correction factor for a day [-]
i sub-time-step index 1…n

The correction of the interpolated global radiation itself (in opposition to the calculation of the correction factor) is done within the evaporation calculation scheme by:

$$RG_{eff} = RG \cdot \left(1 + (1 - cr_0) \cdot SSD \cdot \left[\frac{\cos\hat{\Theta}}{\cos Z} - 1\right]\right) \qquad \textbf{(2.5.11)}$$

with RG_{eff} effective (corrected) global radiation [Wh·m^{-2}]
cr_0 empirical factor for consideration of diffuse shortwave radiation (for the Thur-basin around 0.23, see SCHULLA, 1997)
SSD relative sun shine duration (interpolated, without(!) impacts of topographic shadowing)

Besides the correction of radiation, the azimuth angle Ω_s and the zenith angle Z are used for the analyzing of shadowing of cells by other cells.

2.5.2 Modification of temperature

Using the radiation correction term of equation (2.5.11) it is also possible to modify the air temperature. As parameters will be needed the sunshine duration and an empirical parameter c_t:

$$\begin{aligned} T_{corr} &= T_{obs} + c_t \cdot SSD \cdot \ln\frac{\cos\hat{\Theta}}{\cos Z} && \text{if} \quad 0.2 \le \cos\hat{\Theta} \cdot (\cos Z)^{-1} \le 5.0 \\ T_{corr} &= T_{obs} + c_t \cdot SSD \cdot 1.609 && \text{if} \quad \cos\hat{\Theta} \cdot (\cos Z)^{-1} > 5.0 \\ T_{corr} &= T_{obs} - c_t \cdot SSD \cdot 1.609 && \text{if} \quad \cos\hat{\Theta} \cdot (\cos Z)^{-1} < 0.2 \end{aligned} \qquad \textbf{(2.5.12)}$$

By using the sunshine duration it is taken into account that the radiation correction factor is valid for clear sky conditions only. But differences between cells with different aspects will become smaller with higher cloudiness (resp. shorter sunshine duration). If the sky is fully clouded, no temperature modification is done. The value of c_t should be calibrated or taken from serious measurements of temperatures of plots with different aspects. Usually c_t should be less than 5K.

2.5.3 Control File Settings

The radiation correction and temperature adjustment can be parameterized using the following section in a control file. The section is mandatory. If no radiation correction and temperature adjustment should be done, the first row should contain a value of 0 (switches the module off).

```
[radiation_correction]
1                    # 0=ignore this module, 1 = run the module
$time                # duration of a time step in minutes
2                    # control parameter for radiation correction (see below)
$outpath//$Tcorrgrid # name of the grids with the corrected temperatures
$Writegrid        # Writegrid for corrected temperatures
8                    # factor x for temperature correction x * (-1.6 .... +1.6)
$outpath//$ExpoCorrgrid   # name of the grids with the correction factors for the direct radiation
$Writegrid           # Writegrid
$outpath//$Shapegrid # name of the grids for codes 1 for shadow, 0 for no shadow
$Writegrid           # Writegrid
1                    # interval counter, (3 = all 3 hours a new correction is calculated)
1                   # Spitting of the interval, useful for time step=24 hours (then: split=24, -> #
each hour one correction calculation)
```

with control_parameter:

1 = radiation correction WITH shadow WITHOUT temperature correction

2 = radiation correction WITH shadow WITH temperature correction

3 = radiation correction WITHOUT shadow WITHOUT temperature correction,

4 = radiation correction WITHOUT shadow WITH Temperatur

2.5.4 Computing Relative Sunshine Duration from absolute Values

If there are no relative sunshine duration data are available but only absolute values, the conversion into relative sunshine duration is done automatically during the radiation correction based on the latitude and longitude values provided in the control file and on the actual date. The only pre-requisite for this to work is, that the absolute sunshine duration is provided in minutes per time step and that it is configured as an interpolator with the section name [sunshine_duration_absolute]. If the model recognizes such an interpolated grid, it will automatically switch on the conversion into relative sunshine duration, even if no shadowing and radiation correction is done. The reason for placing the sunshine conversion here is, that all the necessary algorithms needed for sun coordinates, sunrise, sunset etc. are already available.

The conversion uses the sunrise and sunset times for the given center of the basin (longitude and latitude as provided in the control file) as well as the meridian, the given times are related to (e.g. 15° East for Central Europe Time CET) and evaluates the fraction of the actual time step that is between sunrise and sunset. The absolute sunshine duration is then divided by that fraction (with daylight) to compute the relative sunshine duration as a value between 0 and 1.

The algorithm works for all possible locations, time zones, hemispheres, even for polar day and night and also, when the sunrise is after the sunset in the same time step, which can happen for daily time steps when the time zone is shifted strongly and the location is far up north or down south on the planet (very short nights).

2.6 Evapotranspiration

2.6.1 Potential Evapotranspiration after Penman-Monteith

In WaSiM, the potential transpiration from plant leaves, the evaporation from bare soil and the evaporation from interception surfaces (usually leaves, too) will be calculated separately (with separate interception evaporation as an option only). However, the following algorithms are

identical for each type of evaporation. Only the input and/or some parameter estimation vary. Also, for multi-layer vegetation, the algorithms for transpiration and interception evaporation are identical for each layer, but the radiation input will vary.

As a recommendation, the approach after PENMAN-MONTEITH (MONTEITH, 1975; BRUTSAERT, 1982) should be used where possible (however, some other methods are available, see following sections). The potential transpiration after Penman-Monteith is given by:

$$\lambda E = \frac{3.6 \cdot \frac{\Delta}{\gamma_p} \cdot (RN - G) + \frac{\rho \cdot c_p}{\gamma \cdot r_a}(e_s - e) \cdot t_i}{\frac{\Delta}{\gamma_p} + 1 + r_s / r_a} \qquad \textit{(2.6.1)}$$

with λ latent vaporization heat $\lambda = (2500.8 - 2.372 \cdot T)$ KJ·Kg^{-1}, T: temperature in °C
E latent heat flux in mm·m^{-2} ≡ kg·m^{-2} ($\rightarrow$ [λE] = KJ·m^{-2})
Δ tangent of the saturated vapor pressure curve [hPa·K^{-1}] (see eq. (2.6.2))
R_N net radiation, conversion from Wh·m^{-2} to KJ·m^{-2} by a factor 3.6 [Wh·m^{-2}]
G soil heat flux (here: $0.1 \cdot R_N$) [Wh·m^{-2}]
ρ density of dry air = $p/(R_L \cdot T)$ (at 0 °C and 1013,25 hPa: ρ = 1.29 [Kg·m^{-3}]
c_p specific heat capacity of dry air at constant pressure
c_p = 1.005 [KJ·(Kg·K)$^{-1}$]
e_s saturation vapor pressure at the temperature T [hPa] (see eq. (2.6.2))
e actual vapor pressure (observed) [hPa]
t_i number of seconds within a time step
γ_p psychrometric constant [hPa·K^{-1}]
r_s bulk-surface resistance [s·m^{-1}]
r_a bulk-aerodynamic resistance [s·m^{-1}]

Equation (2.6.1) has units of energy flux, i.e. the flux of latent heat in KJ·m^{-2}. The denominator has no units. Both terms of the nominator are given in units of KJ·m^{-2} (1Wh·m^{-2} = 3.6 KJ·m^{-2}). By dividing of λE by the latent heat λ [KJ·Kg^{-1}] the unit changes into Kg·m^{-2}, which is the height of water in mm. The model uses for each land use type monthly values for a minimum bulk-surface resistance. To get the real evapotranspiration ETR, the potential evapotranspiration is then reduced according to the actual soil moisture (see page 75 in chapter 2.8 for real evapotranspiration).

The slope Δ of the vapor pressure function, given in hPa·K^{-1}, is found as first derivative of e.g. the MAGNUS-Formula (TETENS, 1932)

$$e_s = 6.1078 \cdot e^{\frac{17.27 \cdot T}{237.3 + T}} \qquad \textit{(2.6.2)}$$

$$\Delta = \frac{\partial e_s}{\partial T} = \frac{25029}{(237.3 + T)^2} \cdot e^{\frac{17.27 \cdot T}{237.3 + T}}$$

with T air temperature [°C]

The psychrometric-„constant“ γ_p, given in hPa·K^{-1}, is a function of pressure and temperature:

$$\gamma_p = f(p, T) = \frac{c_p \cdot p}{0.622 \cdot \lambda} \qquad \textit{(2.6.3)}$$

with p air pressure (e.g. from barometric formula) [hPa]
0.622 relation of the molecular weights of water vapor and dry air

The air pressure p in (2.6.3) may get form a barometric formula (using an average pressure at sea level):

$$p \approx 1013 \cdot e^{-\frac{h_M}{7991 + 29.33 \cdot Tv}} \qquad \textbf{(2.6.4)}$$

with p air pressure at level h_M [hPa]
h_M altitude [m a.s.l.]
T_v average, virtual temperature of the air column [°C]

The net radiation R_N is the difference of shortwave (R_K) and long wave radiation (R_L). Shortwave direct and diffuse radiation are the global radiation *RG*. The reflected radiation R_{reflex} is estimated using the albedo α:

$$R_N = \underbrace{\left(R_{Sun} + R_{sky} - R_{reflex}\right)}_{R_K} - \underbrace{\left(R_{out} - R_{in}\right)}_{R_L} \qquad \textbf{(2.6.5)}$$

$$R_N = (1 - \alpha) RG - R_L \qquad \textbf{(2.6.6)}$$

with RN net radiation [Wh·m^2]
$\alpha \cdot RG$ reflected shortwave radiation R_{reflex} [Wh·m^2]
α albedo [-]

If no global radiation is observed, it can be estimated using the sunshine duration by:

$$RG = I_{norm} \cdot \left(cr_0 + cr_1 \cdot SSD + cr_2 \cdot SSD^2 + cr_3 \cdot SSD^3\right) \qquad \textbf{(2.6.7)}$$

with I_{norm} theoretical clear sky global radiation [Wh·m^2] after eq. (2.5.7)
SSD relative sunshine duration (ranging from 0 to 1) [-]
$cr_0 \ldots cr_3$ empirical constants [-]
$cr_0 = 0.23$, $cr_1 = 1.77$, $cr_2 = -2.28$, and $cr_3 = 1.28$ for the Thur-basin

I_{norm} is calculated using equations (2.5.6) and (2.5.9) . For daily time steps, I_{norm} is calculated for a number of bright sub-time-steps and averaged for each day.

If no sunshine duration is observed but only global radiation (one of both must be measured), sunshine duration can be modeled using a polynomial relation between sunshine duration and the fraction of observed to theoretical global radiation:

$$SSD_{rel} = cs_0 + cs_1 RG_{rel} + cs_2 RG_{rel}^2 + cs_3 RGrel^3 \qquad \textbf{(2.6.8)}$$

with $cs_0 \ldots cs_3$ polynomial-coefficients ($cs_0 = 0.072$, $cs_1 = -0.808$, $cs_2 = 2.112$, $cs_3 = -0.239$)
RG_{rel} relative global radiation [Wh·m-2]

where

$$RG_{rel} = RG_{mess} / RG_{max} \qquad \textbf{(2.6.9)}$$

with RG_{mess} observed global radiation (or interpolated) [Wh/m^2]

and:

$$RG_{max} = \sum_{i=1}^{n} I_{toa} \cdot \psi_a\left(t_J\right)^{\frac{p(h_M)/p_0}{\cos Z_i}} \cdot \cos Z_i \cdot t_i \qquad \textbf{(2.6.10)}$$

with RG_{max} maximum possible global radiation [Wh·m^{-2}]
I_{toa} incoming shortwave radiation at the top of atmosphere [Wh·m^{-2}] (solar constant ≈ 1368 W·m^{-2})
$\psi_a(t_J)$ extinction coefficient after equation (2.6.11) [-]
$p(h_M)$ air pressure at altitude h_M after equation (2.6.4) [hPa]

p_0 average air pressure at sea level (2.14.233 hPa) [hPa]
n number of sub-time-steps within a day [-]
Z_i zenith angle of the sun for sub-time-step i [rad]
t_i length of sub-time-step i (max. 1 h) [h]

In equation (2.6.1 0) the extinction coefficient is given by:

$$\psi_a(t) = \bar{\psi}_a + 0.5 \cdot \Delta\psi_a \cdot \cos((t_J + 10)/365 \cdot 2 \cdot \pi) \qquad \textbf{(2.6.11)}$$

with $\psi_a(t)$ extinction coefficient for a given time t [-]
$\bar{\psi}_a$ temporal mean extinction coefficient (e.g. 0.88) [-]
$\Delta\psi_a$ range of the extinction coefficient (e.g. 0.1, meaning a range in the above example of 0.83...0.93) [-]
t_J Julian day (1st January = 1, 31st December = 365 resp. 366) [-]

The long wave net radiation is given by:

$$R_L = \Delta t \cdot \sigma T^4 \cdot (0.48 - 0.065\sqrt{e}) \cdot (cr_0 + (1 - cr_0) \cdot SSD) \qquad \text{in } Wh \cdot m^{-2} \qquad \textbf{(2.6.12)}$$

with Δt time step → conversion from power to energy (time related) [h]
cr_0 empirical parameter (Thur: $cr_0 \approx 0.23$) [-]
T air temperature [K]
e actual vapor pressure [mbar]
σ Stefan-Boltzmann-constant $\sigma = 5.67 \cdot 10^{-8}$ $Wm^{-2}K^{-4}$

The aerodynamic resistances r_a are estimated by:

$$r_a = \frac{4.72 \cdot \left(\ln \frac{z}{z_0}\right)^2}{1 + 0.54\,u} \qquad \text{in } s \cdot m^{-1} \qquad \textbf{(2.6.13)}$$

with z sampling elevation above ground [m]
z_0 aerodynamic roughness length as parameter for each land use type $z_0 \approx 0.1 \cdot$ (crop height – shift height d) [m] (note: the control file must contain the value of "crop height – shift height", i.e. the effective height.
u wind speed [$m \cdot s^{-1}$]

For effective crop heights $z_0 > 2$ m r_a is given by:

$$r_a = 25/(1 + 0.54\,u) \qquad \textbf{(2.6.14)}$$

The surface resistances r_s are estimated for the bright day by equation (2.6.15) . Since soil evaporation and transpiration are calculated separately, for each of these evaporation fractions a resistance is needed. Interception evaporation and transpiration are also calculated separately (although this is optional), thus there are at maximum three different surface resistances to deal with.

$$r_s = \frac{1}{\frac{(1-A)}{r_{sc}}} \quad \text{for transpiration resistance}$$

$r_s = 2500$ for LAI or vcf=0

$r_s = 0$ for open water

$r_s = r_{si}$ for interception resitance

$r_s = r_{se}$ for soil evaporation

(2.6.15)

with r_s minimum surface resistance [s·m^{-1}]

r_{sc} minimum surface resistance of the plant if fully supplied with water [s·m^{-1}]

1-A evaporation effective vegetation coverage

$A = f^{LAI \cdot vcf}$ with LAI leaf area index, vcf = vegetated fraction, $f \approx 0.6 \ldots 0.7$

r_{se} surface resistance for bare soil (≈ 150...250 s·m^{-1}, defined in the control file per land use) [s·m^{-1}]

r_{si} surface resistance for interception evaporation (≈ 20...100 s·m^{-1}, defined in the control file per land use) [s·m^{-1}]

LAI leaf area index

vcf vegetation covered fraction

At night the following relations are used:

$$r_s = \frac{1}{\frac{LAI \cdot vcf}{2500}} \quad \text{for transpiration resistance}$$

$r_s = 2500$ for LAI or vcf=0

$r_s = 0$ for open water

$r_s = r_{si}$ for interception resitance

$r_s = r_{se}$ for soil evaporation

(2.6.16)

(using the same symbol as in eq. (2.6.15))

The parameters r_{sc} , r_{se} , r_{si} , z_0, LAI and root depth have a specific annual course, which is given as a series of sampling points in the land use table for each land use. The number of such sampling points is not limited, however it is recommended to use not too much or too less of them (e.g. 4 to 12 values). For each day, the actual value will be interpolated according to the two nearest sampling points (which are given for a specific Julian day each).

In order to consider a time shift in the crops phenology with increasing (or decreasing) altitude, these Julian days or sampling points are shifted to later or former days. An additional parameter AltDep is defined for each sampling point. The default parameter value for the above mentioned parameters is valid for 400 m a.s.l. For all other elevations, the sampling point is shifted as shown in the following example. We assume here, that the sampling points (Julian days) d_1 to d_4 have each an altitude dependency. Sampling points d_1 and d_2 should be moved to later days when the elevation is increasing whereas sampling points d_3 and d_4 should move to earlier Julian days. This may be a simple example for the four most important sampling points (d_1 : start of vegetation, d_2 : fully developed vegetation; d_3 : start and d_4 : end of leave fall (start of dormation).

$$\begin{aligned} d_1 &= d_{1,400} + 0.025 \cdot (h_M - 400) \\ d_2 &= d_{2,400} + 0.025 \cdot (h_M - 400) \\ d_3 &= d_{3,400} - 0.025 \cdot (h_M - 400) \\ d_4 &= d_{4,400} - 0.025 \cdot (h_M - 400) \end{aligned}$$

(2.6.17)

with h_M altitude [m a.s.l.]
d_1, d_2, d_3, d_4 Julian days;
resp. of the shutdown of vegetation
$d_{1,400}$ to $d_{4,400}$ initial values for 400 m a.s.l (examples see SCHULLA, 1997)

This mechanism is used for any number of sampling points – except when using dynamic phenology. Then, the dynamically computed sampling points for vegetation start or other specific periods should not have an altitude correction (since they are calculated based on temperature which is "altitude corrected" already). So the AltDep-parameter should be set to 0 for such special sampling points when using dynamic phenology.

If the model is driven in daily time steps, a subdivision into day and night is automatically performed for the Penman-Monteith approach in order to modify the temperature:

$$\begin{aligned} TP_{24h} &= TP_{day} + TP_{night} \\ EP_{24h} &= EP_{day} + EP_{night} \\ EIP_{24h} &= EIP_{day} + EIP_{night} \end{aligned} \qquad \textit{(2.6.18)}$$

with TP potential transpiration (from plants)
EP potential evaporation
EIP potential interception evaporation

$$TP_{\text{day}} = \frac{\Delta_{\text{day}}\left(R_{N,\text{day}} - G\right) + \rho \cdot c_p \left(e_{s,\text{day}} - \bar{e}\right)}{\Delta_{\text{day}} + \gamma\left(1 + r_{s,\text{day}}/r_a\right)} \cdot d \qquad \textit{(2.6.19 a)}$$

$$TP_{\text{night}} = \frac{\Delta_{\text{night}}\left(R_{N,\text{night}} - G\right) + \rho \cdot c_p \left(e_{s,\text{night}} - \bar{e}\right)}{\Delta_{\text{night}} + \gamma\left(1 + r_{s,\text{night}}/r_a\right)} \cdot d \qquad \textit{(2.6.19 b)}$$

with Δ_{day}; Δ_{night} slope of the saturated vapor pressure curve for temperatures at day (T_{day}) resp. at night (T_{night}) [hPa·K^{-1}]

$R_{N,day}$; $R_{N,night}$ net radiation for day and night, resp. [Wh·m^2]

$R_{N,day} = (1 - \alpha) \cdot RG - R_L$ with R_L after (2.6.12) with $T = T_{day}$ **(2.6.20 a)**

$R_{N,night} = - R_L$ with R_L after (2.6.12) with $T = T_{night}$ **(2.6.20 b)**

$e_{s,day}$; $e_{s,night}$ Saturation vapor pressure as function of T_{day} resp. T_{night} [hPa]

$r_{s,day}$; $r_{s,night}$ surface resistance of the vegetation for the bright day [s·m^{-1}] and for the night after eq. (2.6.15) and eq. (2.6.16) resp.

d relative duration of the bright day (h_{ss}-h_{sr})/24 after eq. (2.6.5) [-]

The other evapotranspiration components EP_{day}, EP_{night}, EIP_{day} and EIP_{night} are calculated using the same equations (2.6.19 a) and (2.6.19 b) but using the r_s-values calculated after equations (2.6.15) and (2.6.16) for the respective evaporation component. Chapter 2.6.3 explains how to avoid physically unrealistic evaporation values.

The temperatures T_{day} and T_{night} are given by:

$$T_{day} = T_{24h} + c_{T,day} \cdot \Delta T \qquad \textit{(2.6.21 a)}$$

$$T_{\text{night}} = T_{24}h + c_{T,\text{night}} \cdot \Delta T \qquad \textit{(2.6.21 b)}$$

$c_{T,Day}$ and $c_{T,Night}$ are calculated as follows:

$$c_{T,\mathrm{day}} = c_T + \Delta c_T \cdot \cos\left(\left(t_J + 10\right)/365 \cdot 2 \cdot \pi\right) \qquad \textbf{(2.6.22 a)}$$

$$c_{T,\mathrm{night}} = 1 - c_{T,\mathrm{day}} \qquad \textbf{(2.6.22 b)}$$

with $c_{T,Day}$ fraction of ΔT, which is added to the daily average T to get T_{day}
$c_{T,Night}$ fraction of ΔT, which is subtracted from the daily average T to get T_{Night}
c_T mean fraction of ΔT for T_{day}
Δc_T range of the fraction c_T during a year (0.1 … 0.15)

The temperature fluctuation range ΔT is given by:

$$\Delta T = f\left(t_J, h_M, SSD\right) = \Delta T_{\mathrm{Sea},t_J} \cdot e^{-h_M/k_T} \cdot SSD \qquad \textbf{(2.6.23)}$$

with $\Delta T_{Sea,tJ}$ temperature range valid for sea level and for Julian day t_J (1 … 365), interpolated using values like from table 2.6.1 on page 71 [K]
k_T recession constant valid for all days of a year [m]
SSD relative sunshine duration as daily value [-]
h_M altitude above sea level [m]

The recession constant k_T was empirically found to be around 1654 m, valid for northern Switzerland. Table 1 contains the monthly values of differences between average day and average night temperatures.

Table 2.6.1: monthly differences between day and night average temperatures, referring to sea level, valid for northern Switzerland

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
ΔT_{Meer} [K]	3.3	4.4	6.1	7.9	9.4	10.0	9.9	9.0	7.8	6.0	4.2	3.2

2.6.2 Evaporation from open Water

A special case is evaporation from open water. Since WaSiM is using the same Penman-Monteith algorithm, the parameter LAI, z_0, r_{sc}, and vcf have to be set at specific values. Since open water should not be used as a land use code. Lakes may grow and shrink, so what land use is to be used for cells belonging to a lake without water in it? → use the land use code as usual, e.g. bare soil or grass and let WaSiM deal with it. The lake model contains a set of parameters which should be used in case the lake cells contain water – regardless of the defined land use. Thus, the parameters defined in the land use table are used when the lake falls dry and the parameters for open water are used when the lake fills up with water. This mechanism works fine for all types of lakes, e.g. small reservoirs which are usually dry and grown with grass, but also for real lakes which never fall dry.

The recommended values for the mentioned parameters for open water are:

$Albedo_{openwater} = 0.1$

$z0_{openwater} = 0.1..0.4$

All the other parameters are set to fix values as follows:

$LAI_{openwater} = 0$

$rsc_{openwater} = 0$

$\mathrm{vcf}_{\mathrm{openwater}} = 0$

Also, the net radiation is calculated independent on LAI and vcf, so those parameters are not used at

all for open water.

Together with the calculation of the surface resistance in equation (2.6.15) and (2.6.16) this makes sure that open water will evaporate with a maximum evaporation rate (*LAI* and *vcf* are unimportant for open water since they are used for the r_{sc} -calculation only)

2.6.3 Energy Balance Aspects for interception evaporation, transpiration and soil evaporation

The fact that there are multiple evaporation components which have to share the available radiation and which can not fill up the saturation deficit more than once but which are calculated independently from each other brings up a conflict. How does WaSiM deal with that conflict?

The solution is a mixture between physically based model assumptions and some empirical approaches.

At first, the soil evaporation can be separated from transpiration and interception evaporation: The parameter *vcf* (vegetated fraction of soil) is used to scale the soil evaporation: the effective soil evaporation is calculated by scaling the result of (2.6.19 a) and (2.6.19 b) (both applied for calculating EP using r_{se} as r_s) by parameter *vcf* . Also, the available radiation will be reduced by possible layers above the ground layer (when using multi-layer vegetation). Vice versa, the interception evaporation and transpiration will be taken from the vegetated area of the cell only. Since this is done in the soil model it is not completely possible to calculate the potential evaporation exactly, especially when using multi-layer vegetation.

Somewhat more complex is the balancing of interception evaporation and transpiration. Here we need some information on the real interception evaporation first. If not all potential interception evaporation could be realized (because of too less or no water on the leafs), then some energy is left for transpiration. This part could not be larger than the potential transpiration itself. So the total amount of (real) interception evaporation and (real) transpiration together cannot be greater than the greater value of TP or EIP. Usually, EIP will be greater than TP (because of the ball-like shape of the water drops in the interception storage there is a much higher saturation vapor pressure and thus a higher potential interception evaporation). The total potential evapotransration (I.e. the sum of potential soil evaporation, potential interception evaporation and potential transpiration) depends on the presence of water in the interception storage:

- if this storage is filled (with an amount >= EIP), the total potential evapotranspiration is the sum of EP and EIP.
- If the interception storage is empty or if there is less water in the storage than TP, the total potential evapotranspiration is the sum of EP and TP.
- If the storage is filled partly, but exceeding the amount of TP, the total potential evaporation is the sum of the potential soil evaporation and the real interception evaporation EIR, which latter value is in this case equivalent to the content of the interception storage

The mechanism is not purely physically based and there is some empiricism in it. Also, it is not really possible to calculate the total potential evapotranspiration, since the single components have a feed back on each other (via real interception evaporation and soil extraction in the soil model). SO the model can output the total real evapotranspiration with some degree of certainty, but no total potential evapotranspiration. To get these values anyway, one should choose the old method 1 for the interception model. Then, only the potential transpiration is used and split between interception evaporation and transpiration. Also, one should use a one-layer-vegetation only with. This value will be much like the traditional ETP-value.

2.6.4 Potential Evapotranspiration after Wendling

The estimation of potential evapotranspiration after both WENDLING (1975) and HAMON (in FEDERER and LASH, 1983) is applicable only if using a time step of one day or larger. The potential evapotranspiration using this approach is then given by:

$$ETP = (RG(1.1-\alpha)+93\cdot f_k)\cdot\frac{T+22}{150(T+123)} \qquad \textbf{(2.6.24)}$$

with RG global radiation in $J\cdot cm^{-2}\cdot d^{-1}$, converted from $Wh\cdot m^{-2}\cdot d^{-1}$ by multiplication with 0.36 $[J\cdot cm^{-2}\cdot d^{-1}]$

α albedo (0 … 1) [-]

f_k empirical correction factor, (coastal regions of northern Germany ≈ 0.6, other flat regions ≈1, value for northern Switzerland found by comparison with Penman-Monteith evaporation: 0.5 [-]

T daily mean value of air temperature [°C]

The global radiation *RG* in the above formula can be derived as described in chapter 2.5 if it is not measured and interpolated.

2.6.5 Potential Evapotranspiration after HAMON

This estimation method after HAMON for potential evapotranspiration uses temperature values only and may be used for time steps of at least one day.

$$ETP = 0.1651\cdot f_i\cdot h_d/12\cdot\frac{216.7\cdot e_s}{T+273.3} \qquad \textbf{(2.6.25)}$$

with f_i empirical factor, monthly values (see table below)

h_d day length [h]

e_s Saturation vapor pressure at temperature T [hPa]

T temperature [°C]

The correction factors f_i contained in equation (2.6.25) are listed in Table 2.6.1.

Table 2.6.1: Monthly correction factors f_i for HAMON-Evapotranspiration (valid for northern Switzerland)

month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct.	Nov	Dec
F_I	0.5	0.6	0.8	1.1	1.2	1.3	1.2	1.1	1.0	0.9	0.7	0.5

2.6.6 Potential Evapotranspiration after HAUDE

The method for calculating the potential evapotranspiration was suggested by Haude (1955). It is an empirical method (as the methods after Wendling and Hamon) and uses water vapor saturation data only (actual and saturation vapor pressure – which latter of course usually is derived from temperature data). The method was developed for Western Germany only – so it's applicability for other regions should first be verified. Similar to Hamon, it uses monthly correction factors which are vegetation type dependent. The method can be applied to daily time steps only. The total potential evapotranspiration is calculated after

$$ETP_{Haude} = k_M \cdot e_s \cdot (1 - RH) \qquad \textbf{(2.6.26)}$$

with k_M empirical factor, monthly values (see table below)
e_s Saturation vapor pressure at temperature T [hPa]
RH relative air humidity ranging from 0 to 1 [-]

The correction factors k_M contained in equation (2.6.26) are listed in Table 2.6.2.

Table 2.6.2: Monthly correction factors k_M for HAUDE-Evapotranspiration (valid for Germany)

month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct.	Nov	Dec
Grass	0.2	0.2	0.25	0.29	0.29	0.28	0.26	0.25	0.23	0.22	0.2	0.2
Corn	0.11	0.11	0.11	0.17	0.21	0.24	0.25	0.26	0.21	0.18	0.11	0.11
Beech	0.01	0	0.04	0.1	0.23	0.28	0.32	0.26	0.17	0.1	0.01	0
Spruce	0.08	0.04	0.14	0.35	0.39	0.34	0.31	0.25	0.2	0.13	0.07	0.05

If Haude evaporation is to be used in a model setup, please note that the monthly correction factors can be given in the control file for one land use only (see next chapter).

2.7 Control file settings for potential evapotranspiration

The section [evapotranspiration] contains most necessary settings for potential evapotranspiration. However, some parameters like leaf area index (LAI) or vegetation coverage factor (vcf) are defined in the land use table. Note that the monthly Haude-correction-factors are given in this section but that they are valid for a single land use only. When using Haude evaporation one should rather use effective parameters for a mixed land use (or switch to Penman-Monteith).

```
[evapotranspiration]
1          # 0=ignore this module, 1 = run the module
$time      # duration of a time step in minutes
1      #  Method: 1=Penman-Monteith, 2=Hamon (daily), 3=Wendling (daily) 4= Haude (daily)
0.5 0.6 0.8  1.0  1.1 1.1 1.2 1.1 1.0 0.9 0.7 0.5  # monthly correction factors for HAMON approach
0.2 0.2 0.21 0.29 0.29 0.28 0.26 0.25 0.22 0.22 0.20 0.20 # kM for Haude, monthly values (Grass)
0.5        # fk -> factor for Wendling-evapotranspiration (only for Method = 3)
$outpath//$ETPgrid    # result grid for pot. evapotranspiration in mm/dt
$Writegrid            # writecode for this grid, see chapter dealing with control file
$outpath//etp_//$grid//.//$year $hour_mean # statisticfile pot. evapo-Transpiration
$outpath//$ETRgrid    # result grid for real evapotranspiration in mm/dt
$Writegrid            # writecode for this grid, see chapter dealing with control file
$outpath//etr_//$grid//.//$year $hour_mean # statistic for real evapotranspiration
$outpath//$EVAPgrid       # result grid for real evapotranspiration in mm/dt
$Writegrid            # writecode for this grid, see chapter dealing with control file
$outpath//evap//$grid//.//$year $hour_mean # statistic for potential evaporation
$outpath//$EVARgrid       # result grid for real evapotranspiration in mm/dt
$Writegrid            # writecode for this grid, see chapter dealing with control file
$outpath//evar//$grid//.//$year $hour_mean # statistic for real evaporation
$outpath//$ETRSgrid    # result grid for real snow evapotranspiration in mm/dt
$Writegrid            # writecode for this grid, see chapter dealing with control file
$outpath//etrs//$grid//.//$year $hour_mean # statistic for real snow evaporation
$outpath//$EIPgrid       # result grid for pot. interception evaporation in mm/dt
$Writegrid            # writecode for this grid, see chapter dealing with control file
$outpath//eip_//$grid//.//$year $hour_mean # statisticfile pot. interception evaporation
$outpath//rgex//$grid//.//$year $hour_mean # statistic for corrected radiation
+0.23 +1.77 -2.28 +1.28 # coefficients c for Polynom of 3rd order RG=c1+c2*SSD+c3*SSD²+c4*SSD³
+0.072 -0.808 +2.112 -0.239 # coef.s x for Polynom of 3rd order SSD=x1+x2*RG+x3*RG²+x4*RG³
0.88 0.05  # Extinktion coefficient for RG-modelling (Phi and dPhi) (summer phi = phi-dphi,
# winter phi=phi+dphi)
1654.0   # recession constant (e-function for recession of the daily temperature amplitude with
#altitude [m])
3.3  4.4  6.1  7.9  9.4  10.0  9.9  9.0  7.8  6.0  4.2  3.2  # monthly values of the max. daily T-
amplitudes (for 0 m.a.s.l)
0.62  0.1 # part of the temperature amplitude (dt), that is added to the mean day-temperature
# (followed by the range of changing within a year ddt) to get the mean temperature of light day
# in the night: mean night temperature is mean day temperature minus (1-dt)*(temp. amplitude)
```

2.8 Calculating real evapotranspiration

2.8.1 WaSiM-version using the Topmodel-approach

This model version uses a relative simple reduction function of potential evaporation to get the real evaporation. The real evaporation is reduced compared to the potential evaporation if the content of the soil moisture storage drops below a specified level η:

$$\begin{aligned} ETR &= ETP \cdot SB / (\eta \cdot SB_{\max}) & SB &< \eta \cdot SB_{\max} \\ ETR &= ETP & SB &\geq \eta \cdot SB_{\max} \end{aligned} \qquad \textit{(2.7.1)}$$

with ETR real evaporation [mm]
ETP potential evaporation [mm]
SB actual content of the soil water storage [mm]
SB_{max} maximum capacity of the soil water storage [mm]
η threshold value for the soil moisture; below the evaporation is reduced compared to the potential evaporation ($\eta \approx 0.6$ after MENZEL, 1997b)

2.8.2 WaSiM-version using the Richards-equation

The more physically based soil moisture modeling within WaSiM version 2 allows another kind of estimating real evaporation by using the relation between the soil water content Θ and the actual capillary pressure (suction) as given by the VAN GENUCHTEN parameters of the actual soil.

Besides a reduction as a result of dry soils ($\Theta(\psi) < \Theta_{\psi g}$) also the impact of too wet soils ($\eta \cdot \Theta_{sat} < \Theta(\psi) < \Theta_{sat}$) is considered.

$$\begin{aligned} ETR_i &= 0 & \Theta(\psi) &< \Theta_{wp} \\ ETR_i &= ETP_i \cdot (\Theta(\psi_i) - \Theta_{wp}) / (\Theta_{\psi_g} - \Theta_{wp}) & \Theta_{wp} &\leq \Theta(\psi) \leq \Theta_{\psi_g} \\ ETR_i &= ETP_i & \Theta_{\psi_g} &\leq \eta \cdot \Theta_{sat} \\ ETR_i &= ETP_i \cdot (\Theta_{sat} - \Theta(\psi_i)) / (\Theta_{sat} - \eta \cdot \Theta_{sat}) & \eta \cdot \Theta_{sat} &< \Theta(\psi) \leq \Theta_{sat} \end{aligned} \qquad \textit{(2.7.2)}$$

with i index of the soil layer (see the soil model description for WaSiM Version 2 using the RICHARDS-equation)
ETR real evaporation [mm]
ETP potential evaporation [mm]
$\Theta(\psi)$ actual relative soil water content at suction ψ [-]
ψ actual suction (capillary pressure) [m]
η maximum relative water content without partly or total anaerobe conditions ($\approx 0.9 \ldots 0.95$)
Θ_{sat} saturation water content of the soil [-]
$\Theta_{\psi g}$ soil water content at a given suction ψ_g
Θ_{wp} water content of the soil at permanent wilting point (ψ = 1.5 MPa ≈ 150 m)

The parameters for equation (2.7.2) are (I) the relative soil water content at the beginning of the reduction of evaporation due to dry soil $\Theta(\psi_g)$ and (II) the relative soil water content at the beginning of reduction due to anaerobe conditions η. Both parameters are given as suction for each crop type in order to make differences between various crop types. This way the reduction of evaporation depends not only on the soil but also on the crop type and it can easily be converted into a time dependent parameter as well.

For multi-layer vegetation the described algorithm is applied for each layer from top to bottom using the specific root depths end root density distributions of each layers land use (as defined in the primary land use table, see following sub-chapter). After extraction of transpiration, the

evaporation is extracted from the uppermost soil layer only.

2.8.3 Parameterization of single and multi-layer vegetation

Since WaSiM version 7.x the seasonal evolution of vegetation is controlled by a two-part landuse table. Compared to the old landuse table the new parametrization scheme allows the parametrization and simulation of multi-layer vegetation. It consists of two parameter tables: (a) the primary **[landuse_table]** containing the parameter values for each vegetation type and (b) the **[multilayer_landuse]** table which links vegetation types provided by the one-layer (primary) [landuse_table] to complex (multi-layer) vegetation structures. The land use codes within the [multilayer_landuse] table have to correspond to the codes of the landuse grid. The two-part land use table is not limited with respect to the number of table entries (landuse codes). It is structured as follows:

```
[multilayer_landuse]
4                # count of multilayer landuse codes
1  urban_areas      { Landuse_Layers = 7, -9999, -9999; k_extinct = 0.3; LAI_scale = 10; }
2  mixed_forest     { Landuse_Layers = 8,     4,     3; k_extinct = 0.3; LAI_scale = 10; }
3  decidous_forest  { Landuse_Layers = 9,     2,     1; k_extinct = 0.3; LAI_scale = 10; }
4  pasture          { Landuse_Layers = 1, -9999, -9999; k_extinct = 0.3; LAI_scale = 10; }


[landuse_table]
7                # number of following land-use codes
#co- name of the
#de  land-use type
#-- ---------------
 1   grass_low {method = VariableDayCount;
       RootDistr       = 1.0;
       TReduWet        = 0.95;
       LimitReduWet    = 0.5;
       HReduDry        = 3.45;
       IntercepCap     = 0.75;
       Juldays         = 15    46    74    105   135   166   196   227   258   288   319   349;
       Albedo          = 0.25  0.25  0.25  0.25  0.25  0.25  0.25  0.25  0.25  0.25  0.25  0.25;
       rsc             = 90    90    75    65    50    55    55    55    60    70    90    90;
       rs_interception = 0.5   0.5   0.5   0.5   0.5   0.5   0.5   0.5   0.5   0.5   0.5   0.5;
       rs_evaporation  = 130   130   130   130   130   130   130   130   130   130   130   130;
       LAI             = 2     2     2     2     3     4     4     4     4     2     2     2;
       z0              = 0.15  0.15  0.15  0.15  0.3   0.4   0.4   0.3   0.3   0.15  0.15  0.15;
       VCF             = 0.95  0.95  0.95  0.95  0.95  0.95  0.95  0.95  0.95  0.95  0.95  0.95;
       RootDepth       = 0.4   0.4   0.4   0.4   0.4   0.4   0.4   0.4   0.4   0.4   0.4   0.4;
       AltDep          = 0.02  0.02  0.02  0.02  0.02  0.02  -0.02 -0.02 -0.02 -0.02 -0.02 -0.02;}
2   grass_high   {method = VariableDayCount; RootDistr = 1.0; ... ; AltDep = 0.025 ... -0.025;}
3   fern         {method = VariableDayCount; RootDistr = 1.0; ... ; AltDep = 0.025 ... -0.025;}
4   shrubbery    {method = VariableDayCount; RootDistr = 1.0; ... ; AltDep = 0.025 ... -0.025;}
7   urban_grass  {method = VariableDayCount; RootDistr = 1.0; ... ; AltDep = 0.025 ... -0.025;}
8   pine         {method = VariableDayCount; RootDistr = 1.0; ... ; AltDep = 0.025 ... -0.025;}
9   birch        {method = VariableDayCount; RootDistr = 1.0; ... ; AltDep = 0.025 ... -0.025;}
```

The parameters in both tables denote as:

```
k_extinct        : extinction coefficient describing the reduction of radiation in multi-layered vegetation [-]
LAI_scale        : empirical parameter scaling the aerodynamic resistance in multi-layered vegetation [-]

RootDistr        : parameter for root density distributio n [-1: konkav, 1: linear, >1: konvex]
TReduWet         : relative theta value for beginning oxygen stress [-]
LimitReduWet     : relative reduction factor of real transpiration for water-saturated soils [-]
HReduDry         : hydraulic head for beginning dryness stress [m]
IntercepCap      : specific thickness of the water layer on the leaves [mm]
Juldays          : Julian days valid for all following rows
Albedo           : albedo (snow free) [0..1]
rsc              : leaf surface resistance [s/m]
rs_interception  : interception surface resistance [s/m]
rs_evaporation   : soil surface resistance (for evaporation only) [s/m]
LAI              : leaf area index [m ² /m ² ]
z0               : aerodynamic roughness length [m]
VCF              : vegetation covered fraction [0..1]
RootDepth        : root depth [m]
AltDep           : shift in temporal vegetation development per metre altitude [Julian day]
```

2.8.3.1 [multilayer_landuse] table

Each row in the [multilayer_landuse] table begins with an entry for a landuse (grid) code. Then it follows the name of the landuse type and a certain number of parameters within curly brackets. The

first entries within the brackets are the vegetation components or landuse layers. They have to be set in particular order. The uppermost layer (or top layer) is the left entry. Then entries to the underlying vegetation components are following. For all landuse types it has to be set the same number of vegetation components. Missing components get the ignorance parameter "-9999". Apart from the layer code settings, two additional parameters, *k_extinct* and *LAI_scale*, can be defined for each landuse type (cf. COUPMODEL in Jansson und Karlberg, 2001).

k_extinct is the light extinction coefficient which reduces the incoming radiation on its way through the layered vegetation (no consideration of diffuse light components). In conformity with the Lambert-Beer-Law the following relations are used:

$$
\begin{aligned}
&for\ i=1:\ R_{use,i}=VCF_i\cdot R_n\cdot\left(1-e^{-k_{extinct}\cdot LAI_i}\right)\\
&for\ i=2:\ R_{use,i}=VCF_i\cdot\left(R_n-R_{use,i-1}\right)\cdot\left(1-e^{-k_{extinct}\cdot LAI_i}\right)\\
&for\ i=3:\ R_{use,i}=VCF_i\cdot\left(R_n-R_{use,i-2}-R_{use,i-1}\right)\cdot\left(1-e^{-k_{extinct}\cdot LAI_i}\right)
\end{aligned}
\qquad \textbf{(2.7.3)}
$$

(analogical procedure for i > 3)

with
- R_{use} available net radiation [Wh/m ^{2}]
- i vegetation component with i=1 as the top level component (e.g. treetop)
- R_n available (total) net radiation (measured or simulated) [Wh/m ^{2}]
- VCF vegetation covered fraction [-]
- LAI leaf area index [m ^{2} /m ^{2}]

LAI_scale is an empirical parameter used for the calculation of the aerodynamic resistances $r_{a,i>1}$ below the treetop (for the vegetation layers 2..n).

$$r_{a,i>1}=r_{a,i=1}+LAI_{scale}\cdot LAI_{kum} \qquad \textbf{(2.7.4)}$$

with
- $r_{a,i=1}$ aerodynamic resistance of the top level vegetation component [s/m]
- LAI_{kum} cumulated leaf area index above the actual vegetation component [m ^{2} /m ^{2}]

Default values for *k_extinct* and *LAI_scale* are 0.3 and 10, respectively. These values are also used in the case of parameter absence (in the control file). As the parameters *k_extinct* and *LAI_scale* significantly affect the evapotranspiration of multi-layer vegetation, they should be carefully calibrated. According to values from literature, *k_extinct* can be between 0.25 and 0.85 (e.g. 0.5 for grass), whereas the values for *LAI_scale* can range from 1 to 30.

2.8.3.2 [landuse_table]

The primary [landuse_table] has been substantially restructured and extended. In contrast to the old landuse table, the new one is no longer limited to a fixed number of "*Julian day*" entries (so far 4 entries). Now the seasonal evolution of vegetation can be predefined with much higher temporal resolution than before. For each "*Julian day*" entry the [landuse_table] requires entries for at least the following parameters: *albedo*, *rsc*, *rs_interception*, *rs_evaporation*, *LAI*, *z0*, *VCF*, *RootDepth* and *AltDep*. This list of temporally varying parameters contains two new resistance parameters, one for the evaporation of intercepted water (*rs_interception)* and the other for the evaporation of water from the soil surface (*rs_evaporation*). Both parameters are optional (see also chapter 4.10.1.2). The dependence of plant growth on altitude (i.e. of temperature differences with altitude) is controlled by the following approach:

$$Juldays_{i,cor}=Juldays_i+AltDep_i\cdot\left(h_M-400\right) \qquad \textbf{(2.7.5)}$$

with $Juldays_i$ actual "Julian day" entry

cor	adjusted for altitude
$AltDep_i$	shift of the actual parameter *Julday* per meter altitude [Julian days]
h_M	altitude above sea level [m]

In addition to the temporally variable parameters, there are also some static parameters which are not linked to the "Julian Day" entries, like *RootDist*, *TReduWet*, *LimitReduWet*, *HReduDry* and *IntercepCap*. For more information see chapter 4.10.1 .

2.8.4 Reducing real transpiration for wet conditions

Apart from the root density distribution function (parameter *RootDist*) and the threshold value for starting dryness stress (*HReduDry*), three new parameters have been added to the [landuse_table]. Two parameters (*TReduWet*, *LimitReduWet*) are defining the reduction potential of real transpiration due to oxygen stress when soil water content reaches saturation (see also Fig. 2.8.4.1). The third parameter (*IntercepCap*), which is defining the water storage capacity on leaves, is an optional parameter. If this parameter is omitted here, then it is used the default parameter from the [interception_model].

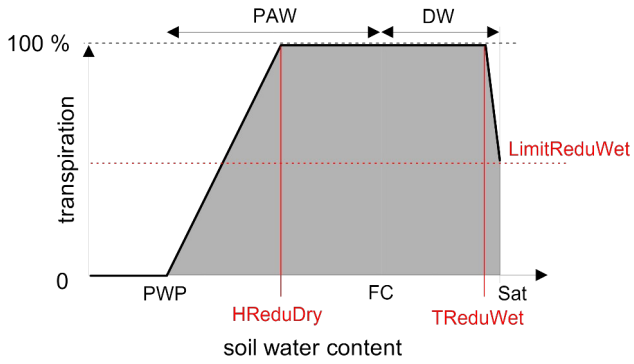


figure 2.8.4.1: Interdependence between transpiration and soil water content. PAW: plant-available water, DW: drainable or gravitational water, PWP: permanent wilting point, FC: field capacity, Sat: soil water content at saturation, HReduDry: threshold value for starting dryness stress, TReduWet: threshold value for starting oxygen stress due to (nearly) water saturated soils, LimitReduWet: maximum reduction of transpiration due to oxygen stress

The extended [landuse_table] allows parameter entries in both (i) multi-line and (ii) single-line formats. Each parameter block (containing all parameter entries of one landuse type) is framed by curly brackets. Parameter groups within one block are separated by semicolons. The method identifier "VariableDayCount" signalized WaSiM that an extended landuse entry has to be read in.

The [landuse_table] can be used in mixed form, i.e. it is possible to use both single-line and multi-line formats within the same table. Moreover, the formats of the old [landuse_table] are furthermore valid. However, for clearity reasons the authors recommend the use of an uniform format of parametrization.

2.9 Snow accumulation and melt

Note 1: the integrated glacier model uses similar but not identical algorithms for snow, firn, and

glacier melt. See the chapter on the dynamic glacier model later in this manual.

Note 2: Since Version 9.06.02 of WaSiM, the snow accumulation and melt algorithms can be applied for unglacierized and glacierized cells in an identical way. Firn and ice melt must be set up specifically for glaciers as usual, but by choosing the new glacier method 13, snow accumulation and snow melt parameters will be taken from the snow model parametrization.
Also, there are some new extensions to the snow model, which are part of the following documentation:

- wind driven snow redistribution
- gravitational snow redistribution by snow slides, that will accumulate snow in lower elevations and on glaciers while ridges and peaks will get rid of too much snow
- new energy balance melt approach, using radiation balance

Note 3: The implementation of snow redistribution algorithms and the new energy balance method was originally done by Michael Warscher in a special WaSiM version (see Warscher et al., 2013). Subsequently, the algorithms were ported to the main WaSiM distribution and tested for several other basins to work with various model configurations. The porting of the snow redistribution algorithms was supported by the Arctic System Science Program of the U.S. National Science Foundation, Award Number 1304905, via the University of Fairbanks

2.9.1 Snow accumulation

The type of precipitation is estimated for each grid cell using the interpolated air temperature during the event. Within a transition range there will be both kinds of precipitation. The fraction of snow is given by:

$$p_{Schnee} = \frac{T_{R/S} + T_{trans} - T}{2 \cdot T_{trans}} \qquad for\ (T_{R/S} - T_{trans}) < T < (T_{R/S} + T_{trans}) \qquad \textbf{(2.9.1)}$$

with p_{Snow} fraction of snow on the total precipitation (0..1)
T air temperature [°C]
$T_{R/S}$ temperature, at which 50 % of precipitation are falling as snow [°C]
T_{trans} ½ of the temperature-transition range from snow to rain [K]

2.9.2 Classic snow melt approaches

- temperature-index-approach

$$M = c_0 \cdot (T - T_{0,m}) \cdot \frac{\Delta t}{24} \qquad for\ T > T_{0,m}\ ,\ else\ M = 0 \qquad \textbf{(2.9.2)}$$

with M melting rate in mm per time step
c_0 temperature dependent melt factor [mm·°C^{-1} ·d^{-1}]
T air temperature, optionally using modification after equation (2.5.12) [°C]
$T_{0,m}$ temperature for beginning with snow melt [°C]
Δt time step [h]

- temperature-wind-index-approach

$$M = (c_1 + c_2 \cdot u) \cdot (T - T_{0,m}) \cdot \frac{\Delta t}{24} \qquad \text{for } T > T_{0,m}\text{, else } M = 0 \qquad \textbf{(2.9.3)}$$

with M melt rate [mm/time step]
c_1 temperature dependent melt factor [mm·°C^{-1} ·d^{-1}]
c_2 wind dependent melt factor [mm·(°C·m·s^{-1} ·d)$^{-1}$]
u wind speed [m·s^{-1}]
T air temperature, casually using modifications after equation (2.5.12) [°C]
$T_{0,m}$ temperature for beginning with snow melt [°C]
Δt time step [h]

- Combination approach after ANDERSON (1973) and extended combination approach after Braun (1985)

This approach was developed for daily time steps. Using it in WaSiM requires a time step of one day or larger. The water equivalent of the snow cover is subdivided into a snow and a water fraction. A coefficient c_1 is used to control the maximum capacity of the water storage as fraction on the total storage (usually around 10 % of the water equivalent). The effective melt is the amount of water exceeding the water holding capacity, where rain and melt are superposed before doing this calculation. Since the storage for the liquid fraction does not necessarily have to be empty at the beginning of a time step, it is required to calculate a refreezing M_{neg} at temperatures below $T_{0,m}$.

$$M_{\text{neg}} = c_{rfr} \cdot RMF \cdot (T - T_{0,m}) \cdot \frac{\Delta t}{24} \qquad \textbf{(2.9.4)}$$

with M_{neg} „negative melt“ (refreezing water) [mm]
c_{rfr} coefficient for refreezing [-]
RMF season dependent radiation melt coefficient [mm·(°C·d)$^{-1}$]
T air temperature, eventually modified after equation (2.5.12) [°C]
$T_{0,m}$ threshold temperature for beginning with snow melt [°C]
Δt time step [h]

After calculating negative melt, the storage's of liquid (S_l) and solid (S_s) fractions on the water equivalent are balanced. The re-frozen amount of liquid water is added to the solid fraction and subtracted from the liquid fraction (limited by the amount of water in the liquid fraction storage).

Melt is calculated only at air temperatures above the threshold temperature $T_{0,m}$. For time steps with precipitation ($P \geq 2$ mm·d^{-1}) melt is composed of radiation melt, melt from sensible heat, melt from latent heat, and melt from energy import from precipitation:

$$M = (M_R + M_S + M_E + M_P) \cdot \frac{\Delta t}{24} \qquad \textbf{(2.9.5a)}$$

with M_R radiation melt [mm]
M_S melt by sensible heat [mm]
M_E melt by latent heat [mm]
M_P melt by import of energy by precipitation [mm]
Δt time step [h]

The melt fractions in equation (2.9.5a) are parameterized by:

$$M_R = 1.2 \cdot T$$
$$M_S = (c_1 + c_2 \cdot u) \cdot (T - T_{0,m}) \qquad \textbf{(2.9.5b)}$$
$$M_E = (c_1 + c_2 \cdot u) \cdot (E - 6.11)/\gamma$$
$$M_P = 0.0125 \cdot P \cdot T$$

with T air temperature [°C]
$T_{0,m}$ threshold temperature for melt [°C]
c_1 temperature dependent melt factor [mm·(°C·d)$^{-1}$]
c_2 wind dependent melt factor [mm·(°C·m/s·d)$^{-1}$]
u wind speed [m·s^{-1}]
γ psychrometric constant, see equation (19) [hPa·K^{-1}]
E saturation vapor pressure at air temperature T [hPa]
(extended approach after Braun, 1985: observed vapor pressure e)
P precipitation [mm]

In a time step without precipitation melt is calculated like in equation (2.9.2) by:

$$M = RMF \cdot (T - T_{0,m}) \cdot \frac{\Delta t}{24} \qquad \textbf{(2.9.6)}$$

with M melt [mm]
RMF radiation melt factor [mm·(°C·d)$^{-1}$]
T air temperature [°C]
$T_{0,m}$ threshold temperature for melt [°C]
Δt time step [h]

2.9.3 Lateral, wind driven snow redistribution

Starting with WaSiM version 9.06.02 there is a new possibility of lateral snow redistribution, which is in fact a kind of snow precipitation correction. It is based on the algorithms described by Warscher et al. (2013), who used ideas of Winstral & Marks (2002), but adopted the algorithms to be applicable to any basin without accounting for turbulences and vertical snow redistribution (which is done here in another approach, see below). The basic principle is to look for cells that are sheltered from the main wind direction and let the snow fall increase there and, on the contrary, decrease the snow fall on wind exposed cells. Which cells are exposed or sheltered is estimated by the directed sky view factor. This is a sky view factor which is not computed for the full circle but for a sector only. If, as an example, the main wind direction is SW, then the directed sky view factor can be computed for the range of 180° to 270°. The generation of such a grid can be done using the latest version of Tanalys.

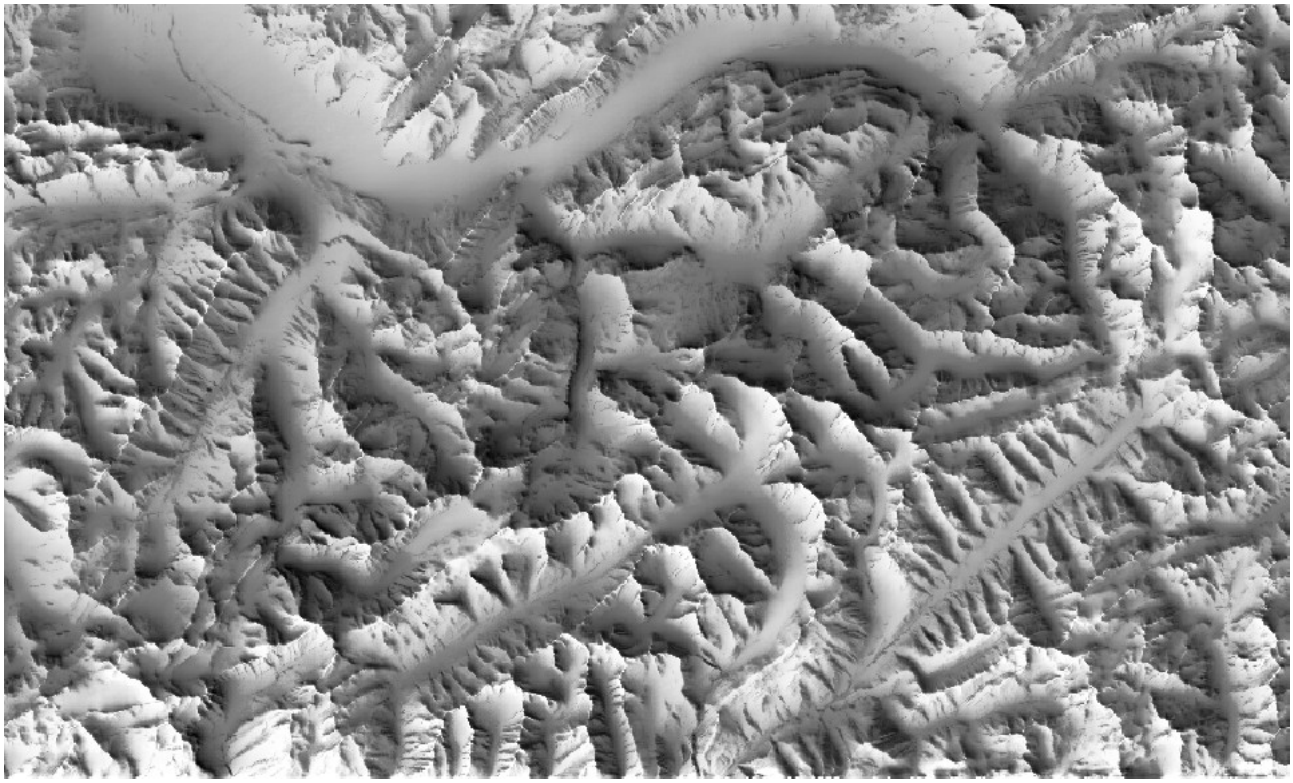


figure 2.9.3.1: Directed sky view factor (SVFdir) for 180° to 270° (i.e. SW) for the central Swiss Alps

Figure 2.9.3.1 shows an example of such a directed sky view factor. It is, by the way, an excellent method of creating a diffusely illuminated terrain model. White stands for cells that are exposed to the main wind direction, black cells are sheltered from that main wind direction. However, the range of this directed sky view factor SVF_{dir} is from 0 to 1 with an average in-between (here: around 0.75). To convert this into the wind correction factor for snow, SVF_{dir} is scaled as follows:

$$h_{eff} = \left(h - h_{min} + h_{offs}\right) / \left(h_{max} - h_{min} + h_{offs}\right) \qquad \textbf{(2.9.7)}$$

with
- h_{eff} effective elevation of the cell
- h actual elevation of the cell
- h_{min} minimum elevation of all cells in the grid
- h_{max} maximum elevation of all cells in the grid
- h_{offs} offset elevation to pronounce snow fall even in lower elevations
  recommended values: 0...1000 (much depending on actual elevation range!)

$$c_{wind} = h_{eff} \cdot \left(D_{max} \cdot \left(1.0 - SVF_{dir}\right) - 1.0\right) + c_{min} \qquad \textbf{(2.9.8)}$$

with
- h_{eff} the result of equation (2.9.7)
- D_{max} maximum deposition, a scaling factor which should be set to values around 2...3. This factor allows for calibrating the average snow correction factor. Higher values will result in higher avarages of c_{wind}.
  Recommended values: 2... **3** ...4
- SVF_{dir} directed sky view factor. Can be calculated by using tanalys (see description of the help tools in chapter 5.1.27 , TANALYS)
- c_{min} minimum correction factor. This value shifts c_{wind} (also the average!). It should be used to make sure that a certain amount of snow will stay on every cell.
  Recommended values: 0.1... **0.2** ...0.3

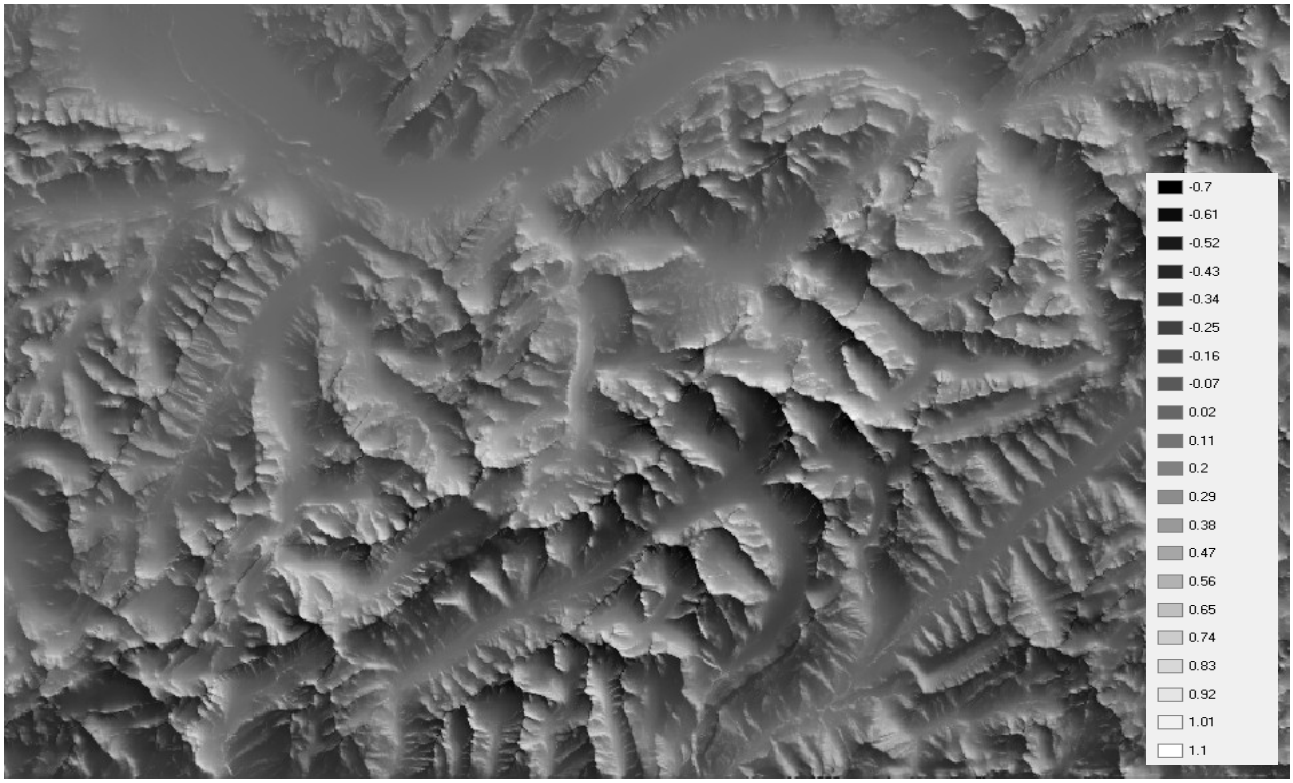


figure 2.9.3.2: wind driven snow correction factors c_wind for the central Swiss Alps region. Average: 0.028, min/max:-0.7, 1.04; profile A-B: see figure 2.9.3.3

As can be seen in Figure 2.3.9.2, the effective wind correction results in a similar, but differently distributed pattern. The average wind correction factor is around 0.028 in this example, thus resulting in around 3% snow correction (this may be way too less, but by changing D_{max} in equation (2.9.8) , the overall average can be moved.

The resulting snow precipitation is calculated by

$$P_{snow} = P_{snow} \cdot (1.0 + c_{wind}) \qquad \textbf{(2.9.9)}$$

with P_{snow} effective snow precipitation

Attention: This correction is applied within the snow model only. If a snow correction was already done in [precipitation_correction], this corrected value is again scaled here. So it may be a good idea to choose values of 1.0 and 0 resp. for the normal precipitation correction for snow in order to make sure that the correction is not too high. For computing balances, the statistics for snow and rain should be used from the snow model rather than the result of the precipitation interpolation, since that value may be too low after applying the snow correction as described here.

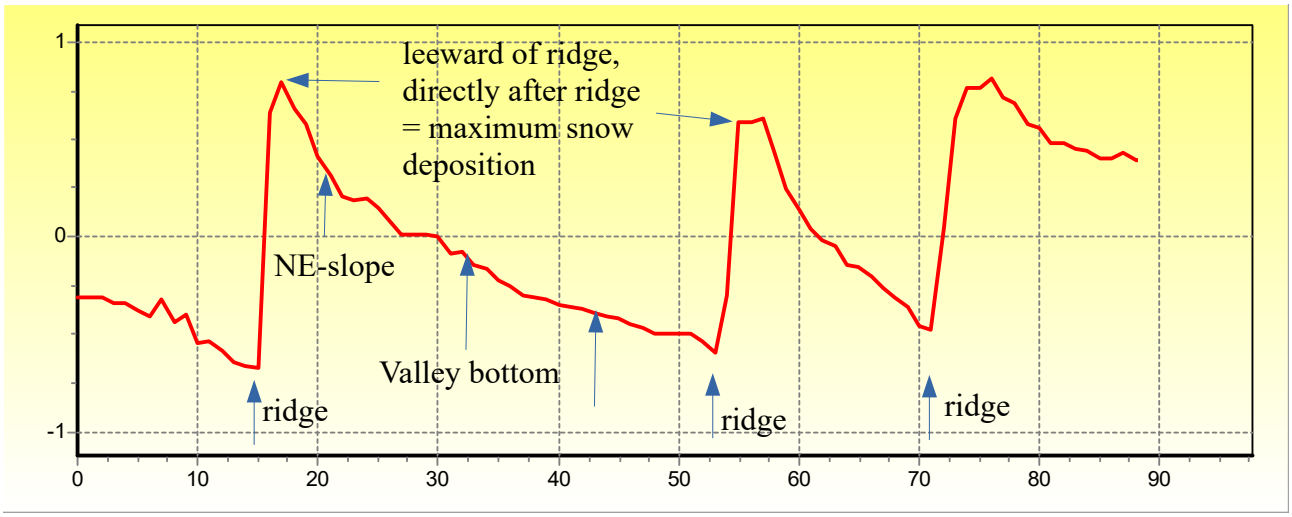


figure 2.9.3.3 profile A-B (see figure 2.9.3.2) for a SW main wind direction

A profile along line A-B through three ridges (line is shown shown in Fehler: Verweis nicht gefunden in blue) is shown in Fehler: Verweis nicht gefunden. The snow correction factor drops to values of about 0.4...0.5 directly before the ridge on the windward side. Directly on the ridge it is lowest, rising to the highest values shortly behind the ridge in leeward direction. From there, it decreases to near zero (= no decrease and no increase) at the valley bottom and decreases further to 0.4...0.5 again when reaching the windward slope of the next mountain reach.

2.9.4 Gravitational snow redistribution

The lateral transport by gravitational snow slides is an important process in mountainous hydrology. Snow sliding from steep slopes of a mountain to lower and not so steep areas do significantly impact glacier mass balances and also the spatial and temporal pattern of snow melt. The algorithms used here where implemented in a special WaSiM version by Warscher et al. (2013) and ported to the common WaSiM version in 2014. They are based on the approach of Gruber (2007). The basic assumption is, that snow can flow as slides to any neighboring cell that has a lower elevation than the actual cell, as soon as the slope to that cell is steep enough. The algorithm is mass conserving, but since it is possible for slides to flow in all four main directions it may happen that snow can also flow to another sub basin. That can change the water balance of a simulation for the two subbasins, but not for the total basin. Once the snow is starting to slide, it may travel not only to the neighboring cell(s), but also from there to other cells, until the slope becomes too low. Thus, the algorithm must start at the highest elevations in order to track moving snow masses.

To compute the mass of snow that flows into neighboring cells, four parameter grids that specify the sliding fractions for each of the main directions (N, E, S, W). These grids can be created using the tool Tanalys (see chapter 5.1.27). The flow from each cell to its neighbors is given by

$$F_{NB} = M_{out} \cdot f_{nb} \qquad \textbf{(2.9.10)}$$

with F_{NB} flow to neighbor NB (1=N, 2=E, 3=W, 4=S)
f_{NB} fraction of flow into the actual direction ($f_1+f_2+f_3+f_4 = 1.0$)
M_{out} total mass of snow that flows from that cell [mm SWE]

The total outflow M_{out} of a cell is determined by

$$M_{out} = \begin{bmatrix} f_{erosion} \cdot SWE + M_{in} - D_{grav} & \text{if } i \geq i_{erosion} \text{ and } SWE > 0 \\ M_{in} - D_{grav} & \text{if } i \geq i_{erosion} \text{ and } SWE = 0 \\ M_{in} - D_{grav} & \text{if } i < i_{erosion} \end{bmatrix} \quad \textbf{(2.9.11)}$$

with M_{in} inflowing mass from all other neighboring cells [mm]
SWE snow water equivalent in the actual cell [mm]
$f_{erosion}$ erosion factor, depending on model time step (in 1/1 of the actual SWE)
i local slope
$i_{erosion}$ lower inclination limit for snow erosion [°], ranging from 0...90°, best values between 30°...55°, depending on spatial resolution
D_{grav} deposition in the actual cell [mm]

The actual deposition of snow in a cell D_{grav} is defined by

$$D_{grav} = \begin{Bmatrix} M_{in} & \text{if } M_{in} < D_{max,grav} \\ D_{max,grav} & \text{if } M_{in} \geq D_{max,grav} \end{Bmatrix} \quad \textbf{(2.9.12)}$$

$$D_{max,grav} = \begin{Bmatrix} \left(1 - \frac{i}{i_{lim}} \cdot D_{lim}\right) & \text{if } i < i_{lim} \\ 0 & \text{if } i \geq i_{lim} \end{Bmatrix} \quad \textbf{(2.9.13)}$$

with D_{lim} upper deposition mass limit [mm] for snow that flows across the cell-by-cell
i_{lim} upper deposition slope limit [°], for steeper slopes, all inflowing masses will be transported to the next downslope cell(s)

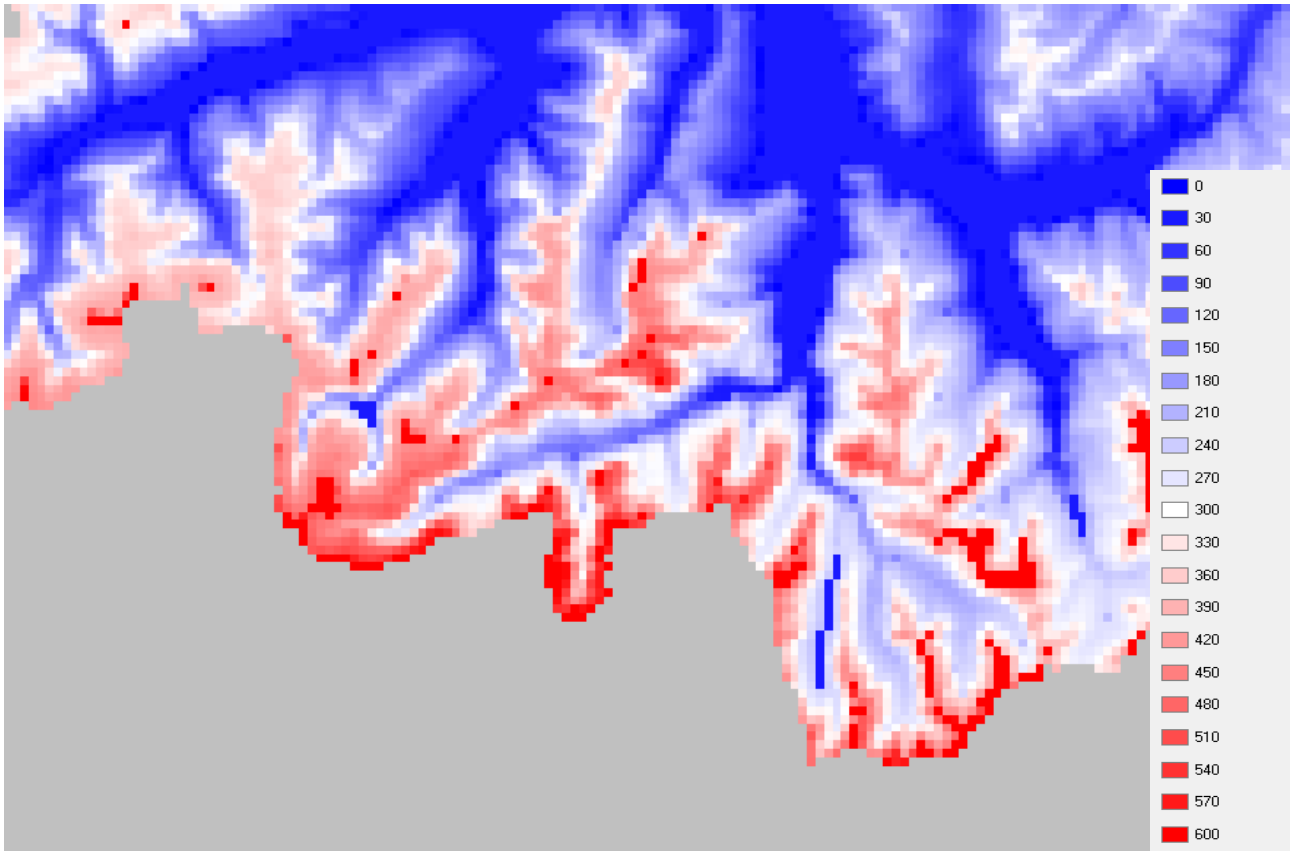


figure 2.9.4.1: snow distribution pattern without slides (in mm SWE)

The effect of lateral snow distribution by gravitational snow slides can be seen by comparing figures 2.9.4.1 and 2.9.4.2: While in figure 2.9.4.1 the distribution is more or less evenly increasing with elevation, in figure 2.9.4.2 there are some regions with almost no snow at steep slopes and a lot of snow in the valley line. This example is for demonstration purposes only, the values of i_{lim} and D_{lim} as well as $i_{erosion}$ and $f_{erosion}$ should be calibrated depending on the spatial and temporal resolution. The algorithm was originally developed for spatial resolutions of 10...50m, but since WaSiM can and will be applied to larger regions with resolutions of e.g. 500m, these parameters may become effective parameters. For instance, the minimum slope for erosion $i_{erosion}$ will get smaller if the spatial resolution gets coarser and the input slope is based on that coarser resolution.

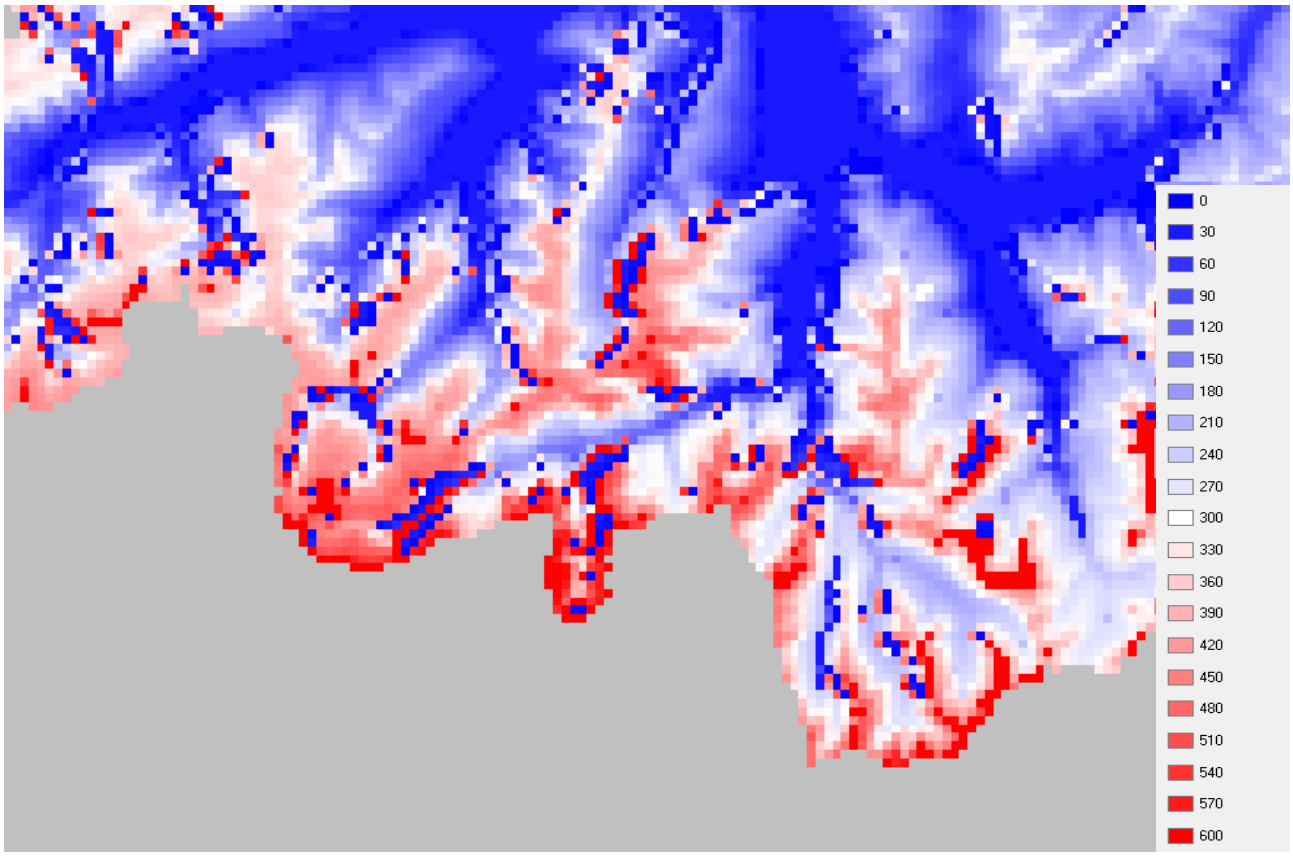


figure 2.9.4.2: snow distribution pattern with slides (in mm SWE)

2.9.5 Energy balance approach for snow melt

Like wind driven and gravitational snow redistribution, the energy balance approach for snow melt was implemented by Warscher et al. (2013) into a special WaSiM version and ported to the common release version by Schulla in 2014. A somewhat more detailed description of the energy balance method can be found in Warscher et al. (2013). Here, only a few key features will be pointed out:

The energy balance approach considers all energy fluxes into and out of the snow pack:

$$Q+H+E+A+G+M_{ae}=0 \qquad \textbf{(2.9.14)}$$

with
- Q short wave and long wave radiation balance-grid [W/m²]
- H sensible heat flux [W/m²]
- E latent heat flux [W/m²]
- A advective heat flux by precipitation [W/m²]
- G ground heat flux [W/m²]; here taken as 2 W/m²

M_{ae} energy available for melting and/or sublimation

The radiation balance is calculated by

$$Q=(1-\alpha)\cdot RG+LW_{\text{in}}+LW_{\text{out}} \quad \textbf{(2.9.15)}$$

with
- α Albedo
- RG incoming shortwave radiation (global radiation) [W/m ^{2}]
- LW_{in} incoming longwave radiation [W/m ^{2}]
- LW_{out} outgoing longwave radiation [W/m ^{2}]

$$LW_{\text{in}}=\sigma\cdot c_{\text{LW,in}}\cdot T_{air}^4\cdot(0.61+0.05\sqrt{e})\cdot(1+c_b\cdot CL^2) \quad \textbf{(2.9.16)}$$

with
- σ Stephan-Boltzmann constant 0.567 ·10 $^{-8}$ W/(m ^{2} K ^{4})
- $c_{\text{LW,in}}$ correction factor (model parameter)
- T_{air} air temperature [K]
- e vapour pressure [Pa]; with different values for surface temperatures above or below 0°C
- c_b factor accounting for cloudiness: 0.24 if precipitation; 0.08 for cloudiness <0.2, 0.17 for cloudiness between 0.2 and 0.6, 0.2 for cloudiness > 0.6
- CL cloudiness [-] (is calculated in evaporation module depending on global radiation after $0.997+0.4473\cdot RG_{\text{rel}}-1.4059\cdot RG_{\text{rel}}^2$)

$$LW_{\text{out}}=-\sigma\cdot c_{\text{LW,out}}\cdot T_{surf}^4 \quad \textbf{(2.9.17)}$$

with
- σ Stephan-Boltzmann constant 0.567 ·10 $^{-8}$ W/(m ^{2} K ^{4})
- $c_{\text{LW,out}}$ correction factor (model parameter)
- T_{surf} surface temperature [K] (is 273,16K for T_{air} > or = 0 and snow pack temperature for T_{air} < 0

Sensible heat flux H and latent heat flux E are calculated by

$$H=18.85\cdot(0.18+0.098\cdot u)\cdot(T_{air}-T_{surf}) \quad \textbf{(2.9.18)}$$

with
- u wind speed [m/s]

$$E=32.82\cdot(0.18+0.098\cdot u)\cdot(e_{air}-e_{snow}) \quad \textbf{(2.9.19)}$$

with
- u wind speed [m/s]
- e_{air} vapor pressure in the air for T_{air}
- e_{snow} vapor pressure over snow for T_{surf}

The advective heat flux (in case of precipitation) is defined by

$$A=(c_{lw}\cdot(T_{air}-273.16)\cdot P_{rain})+(c_{ice}\cdot(T_{air}-273.16))\cdot P_{snow} \quad \textbf{(2.9.20)}$$

with
- P_{rain} amount of liquid precipitation (rain)
- P_{snow} amount of solid precipitation (snow)
- c_{lw} specific heat capacity of liquid water [J/Kg/K] (used value: 4180)
- c_{ice} specific heat capacity of frozen water [J/Kg/K] (used value: 2100)

If the energy balance is positive, the remaining energy M_{ae} can be either used for snow melt or for sublimation. To decide, which process will be favored, the so called wet bulb temperature (the temperature that results from evaporating as much water as is needed until the air is saturated) is calculated by

$$e_{air} = e_{sat}(T_{wet}) - \gamma \cdot (T_{air} - T_{wet}) \cdot p \cdot (1.0 + 0.00115 \cdot T_{wet}) \quad \textbf{(2.9.21)}$$

with e_{air} vapor pressure in the air

$e_{sat}(T_{wet})$ saturation vapor pressure at wet bulb temperature T_{wet}

γ psychrometric constant (0.00066 K^{-1})

p air pressure [Pa] (is taken from the barometric elevation formula as an approximation)

Equation (2.9.21) must be solved iteratively, since T_{wet} is only implicitly available. The first guess for T_{wet} is taken from

$$T_{wet,1} = (T_{dew} + 2 \cdot T_{air})/3.0 \quad \textbf{(2.9.22)}$$

with T_{dew} dewpoint temperature

$$T_{dew} = 272.62 * F/(22.46 - F) \quad \textbf{(2.9.23)}$$

$$F = \log((x \cdot p)/((0.622 + x) \cdot 6.112)) \quad \textbf{(2.9.24)}$$

$$x = 0.622 * e_{air}/(p - e_{air}) \quad \textbf{(2.9.25)}$$

with x mixing ratio of water and air masses

After solving the iteration (when T_{wet} between two iterations differs no more than 0.025K), the decision can be made if the energy balance remainder M_{ae} can be used for sublimation (if T_{wet} <= 0°C) or for snow melt (T_{wet} > 0°C)

Snow melt:

$$M = (T_{snow} - 273.16) \cdot c_{ice} \cdot SWE/337500.0 \quad \textbf{(2.9.26)}$$

Snow sublimation (will be added to evaporation)

$$E_s = (T_{snow} - 273.16) \cdot c_{ice} \cdot SWE/2835500.0 \quad \textbf{(2.9.27)}$$

Note that the conversion to the amount of melted resp. sublimated water differs by a factor of approximately 7, since the energy needed for melting (337.5 KJ/Kg) is only 1/7th of the energy required for sublimation (2835.5 KJ/Kg). So if the wet bulb temperature is below 0°C, which is the case at very dry conditions only, the snow cover will be relatively stable and evaporates even at high temperatures (up to 7 or 8°C) with a rate of only 1/7th compared to the melting rate at the same temperature but at more moist conditions.

2.9.6 Control file settings for snow model

The parameters of the snow model are globally defined. There is no possibility to set different parameters for different land uses or different sub basins yet.

```
# snow model methods:
# conventional:
# 1 = T-index
# 2 = T-u-Index
# 3 = energy balance approach after Anderson
# 4 = extended energy balance approach after Braun (based on Anderson) with observed vapor pressure
# extended methods (introduced by Michael Warscher in 2013, implemented into WaSiMs
# main distribution by J. Schulla, 2014):
# 5 = Enhanced energy balance approach (with more sophisticated parameter estimations)
# 6 = enhanced Energy balance + gravitational slides
# 7 = enhanced Energy balance + gravitational slides + wind redistribution
# 8 = T-Index + gravitational slides
# 9 = T-Index + gravitational slides + wind redistribution
# 10 = T-Index + wind redistribution
# 11 = enhanced Energy balance + wind redistribution
# 12 = T-u-Index + gravitational slides
```

```
# 13 = T-u-Index + wind redistribution
# 14 = T-u-Index + gravitational slides + wind redistribution

[snow_model]
1                   # 0=ignore this module, 1 = run the module
$time               # duration of a time step in minutes
1                   # method, see comments above the section header
1.0                 # transient zone for rain-snow (T0R +- this range)
0.2                 # T0R    temperature limit for rain (Grad Celsius)
0.2                 # T0     temperature limit snow melt
0.1                 # CWH    storage capacity of the snow for water (relative part)
1.0                 # CRFR   coefficient for refreezing
1.65                # C0     degree-day-factor mm/d/C
0.8                 # C1     degree-day-factor without wind consideration  mm/(d*C)
0.7                 # C2     degree-day-factor considering wind mm/(d*C*m/s)
0.07                # z0     roughness length cm for energy balance methods (not used)
1.3                 # RMFMIN minimum radiation melt factor      mm/d/C comb. method
2.3                 # RMFMAX maximum radiation melt factor      mm/d/C comb. method
0.40                # Albedo for snow (Min)
0.85                # Albedo for snow (Max)
$outpath//$rain_rate         # rain rate
$Writegrid                   # 0, if no grid-output is needed, else one of the codes described above
$outpath//rain//$grid//.//$year $hour_mean # rain rate
$outpath//$snow_rate         # snow rate
$Writegrid                   # 0, if no grid-output is needed, else one of the codes described above
$outpath//snow//$grid//.//$year $hour_mean # snow rate
$outpath//$days_snow         # days with snow (SWE > 5mm)
$Writegrid                # 0, if no grid-output is needed, else one of the codes described above
$outpath//sday//$grid//.//$year $hour_mean # days with snow (SWE > 5mm)
$outpath//$snow_age          # snow age (days without new snow)
$Writegrid                # 0, if no grid-output is needed, else one of the codes described above
$outpath//sage//$grid//.//$year $hour_mean # days since last snowfall
$outpath//albe//$grid//.//$year $hour_mean # Albedo
$outpath//$snowcover_outflow # discharge from snow, input (precipitation) for following modules
$Writegrid                   # 0, if no grid-output is needed, else one of the codes described above
$outpath//qsch//$grid//.//$year $hour_mean # melt flow (or rain, if there is no snow cover) in mm/dt
$outpath//$melt_from_snowcover # discharge from snow, input (precipitation) for following modules
$Writegrid                   # 0, if no grid-output is needed, else one of the codes described above
$outpath//qsme//$grid//.//$year $hour_mean # melt flow in mm/dt
$outpath//$SSNOgrid        # name of the grids with the snow storage solid in mm
$Writegrid                   # 0, if no grid-output is needed, else one of the codes described above
$outpath//$SLIQgrid          # name of the grids with the snow storage liquid in mm
$Writegrid                   # 0, if no grid-output is needed, else one of the codes described above
$outpath//ssto//$grid//.//$year $hour_mean # total snow storage, in mm, (liquid and solid fraction)
$outpath//$SSTOgrid          # name of the grids with the total snow storage solid AND liquid in mm
$Writegrid                   # 0, if no grid-output is needed, else one of the codes described above
$readgrids                   # 1=read snow storage solid, liquid grids from disk, 0=generate new grids
# now some new parameters and file names follow for the snow model extensions implemented in 2014 (originally done
# by M. Warscher and adopted and a little bit extended to the publicly available release in 2014 by J. Schulla)
$outpath//$snowtemperaturgrid   # result grid with snow pack temperature
1//$Writegrid               # 0, if no grid-output is needed, else one of the codes described above
$outpath//snowtemp//$grid//.//$code//$year $once_per_interval # temperature of the snow pack (energy balance model)
$outpath//$snowsurftemperaturgrid  # result grid with
1//$Writegrid               # 0, if no grid-output is needed, else one of the codes described above
$outpath//ssrftemp//$grid//.//$code//$year $once_per_interval # temperature of the snow surface (energy bal. Model)
$outpath//$inputmassgrid    # result grid with snow surface temperature
1//$Writegrid               # 0, if no grid-output is needed, else one of the codes described above
$outpath//inpmass//$grid//.//$code//$year $once_per_interval # input mass for snow slides
$outpath//$mobilemassgrid   # result grid with input mass for gravitational snow slides
1//$Writegrid               # 0, if no grid-output is needed, else one of the codes described above
$outpath//mobmass//$grid//.//$code//$year $once_per_interval # moving/mobile mass for snow slides
$outpath//$depositiongrid   # result grid with deposition amount for each cell (gravitational snow slides)
1//$Writegrid               # 0, if no grid-output is needed, else one of the codes described above
$outpath//deposit//$grid//.//$code//$year $once_per_interval # deposition amount (mm) for gravitational slides
55                       # maximum deposition slope (0...90)
2                        # scaling for maximum deposition (0..inf)
30                       # minimum slope for creating slides (0...90) (scale dependent!)
0.01                     # fraction of snow pack that forms the slide (0...1]
1.00    # LWINcorr: correction factor for incoming long wave radiation for fine tuning the energy balance (accouting
#         together with LWOUTcorr for errors in cloudiness and albedo); recommended Values: 0.8...1.2
0.80    # 0.85 # LWOUTcorr: correction factor for outgoing long wave radiation for fine tuning the energy balance
#       (accouting together with LWINcorr for errors in cloudiness and albedo): recommended Values: 0.8...1.2
```

2.9.7 Output of Pure Snow Melt

Pure snow melt was already defined in the control file, usually as qsme:

```
[snow_model]
1                     # 0=ignore this module, 1 = run the module
...
$outpath//$snowcover_outflow # discharge from snow statistics (rain and melt)
1//$Writegrid
```

```
$outpath//qsch//$grid//.//$code//$year $hour_mean # melt flow and/or rain
$outpath//$melt_from_snowcover # melt flow only statistics
1//$Writegrid
$outpath//qsme//$grid//.//$code//$year $hour_mean # melt flow only in mm/dt
```

However, WaSiM did output the same values for both grids and statistics, i.e. there was only the snow cover outflow including rain. The new versions (starting from 10.08.00) only writes the pure snow melt into the qsme grid and statistics.

Note: This does not work when using the layered snow model. There, qsme will always be the total outflow from the snow model, since it's not easily possible by design to clearly define what's snow melt and what's rain. The snow pack is simulated as part of the soil there and melting snow is reducing the layer thicknesses and flowing to the next layer below, where it is no longer possible to differentiate between rain and snow melt (also because there may be a mixing of both components during the passage of melt water from upper layers to lower layers with melt water from other layers and/or infiltrating rain).

2.9.8 Time variable snow melt coefficients for all T-index and T-u-index approaches

2.9.8.1 Introduction

Another 2020 extension is the possibility to provide 12 monthly values instead of one constant value for the snow melt coefficients c0, c1 and c2 (c0 = degree day factor for T-index approach; c1 = degree day factor for T-u-index approach, c2 = wind dependend factor for T-u-index approach).

Background: Simple snow melt approaches with only the temperature and, if available, the wind speed as input variables, are still widely in use since high quality and high resolution (of both, space and time) data for energy balance approaches are often not available. Regarding the complexity of the built-up and melt-down of a snow cover and it's dependency on factors like humidity, net radiation (short wave and long wave), wind speed, temperature and also snow pack properties (density, pore volume, stratification) it becomes clear that an approach with only one or two parameters can not always estimate the snow melt for all cells of a larger basin for all days of a year in comparably good quality. To give the model user a little bit more flexibility, this new developed feature offers at least the possibility to adjust melting rates for seasonal effects.

The key features are:

- all T-index snow melt methods and T-u-index snow melt methods can use 12 monthly paramaters instead of a single constant parameter (snow melt methods 1, 2, 8, 9, 10, 12, 13 and 14)
- the parameters are valid for the 16-th day of a month (00:00), between those days, the actual parameters are interpolated

2.9.8.2 Requirements in detail

- To use 12 monthly values for snow melt parameters, one of the T-index or T-u-index melt methods must be used:
  - 1: T-index
  - 2: T-u-index
  - 8: T-Index + gravitational slides
  - 9: T-Index + gravitational slides + wind redistribution
  - 10: T-Index + wind redistribution
  - 12: T-u-Index + gravitational slides

- 13: T-u-Index + wind redistribution
- 14: T-u-Index + gravitational slides + wind redistribution

- The control file must contain 12 monthly parameter values where it usually looks for one value only:

  sample control file

```
[snow_model]
1          # 0=ignore this module, 1 = run the module
$time      # duration of a time step in minutes
14         # method 1=T-index, 2=t-u-index, 3=Anderson comb., 4=extended com.
1.0        # transient zone for rain-snow (T0R +- this range)
0.85       # T0R    temperature limit for rain (Grad Celsius)
0.85       # T0     temperature limit snow melt
0.15       # CWH    storage capacity of the snow for water (relative part)
0.5        # CRFR   coefficient for refreezing
1.4  1.4  1.4  1.6  1.8  1.9  1.9  1.8  1.8  1.8  1.4  1.4 # C0 ddf mm/d/C
0.7  0.7  0.7  0.8  1.0  1.1  1.1  1.0  1.0  1.0  0.7  0.7 # C1 ddf mm/(d*C)
0.4  0.4  0.4  0.6  0.7  0.8  0.8  0.7  0.7  0.7  0.4  0.4 # C2 ddf mm/(d*m/s)
0.07       # z0     roughness length cm for energy bilance methods (not used)
3.5        # RMFMIN minimum radiation melt factor      mm/d/C comb. Method
…
<snip>
```

- If WaSiM doesn't find at least 12 values for the chosen method, it will use the first value as constant parameter for the entire model period (default for backwards compatibility)
- If there are more than 12 values given, only the first 12 values will be read
- the values are valid for the 16th day of each month. For all other time steps, values will be interpolated between the last and the next parameter. For days before January 16, the 12 th value will be used for interpolation between December 16 and the actual day. Also, for days after December 16, the first parameter for January 16 will be used for interpolation.
- All other parameters of the snow model are still constant, single values, valid for the entire model run.

2.9.8.3 Sample results

Figure 2.9.8.1 shows a comparison of observed runoff and modeled runoff for the old default method (blue) and the new variable method (green). Since the time series are overlapping strongly, the following figures 2.9.8.2 and Fehler: Verweis nicht gefunden show both model results separately. As can be seen in figure 2.9.8.2, the modeled runoff is too high in March and the first half of April, whereas it is too low in May and June. By setting the T-u-index parameters like in the above sample control file snippet, the overall model quality could significantly improved, see figure Fehler: Verweis nicht gefunden: modeled runoffs in March and April are smaller, whereas the runoffs in May and June are higher than in the original model run with constant parameters.

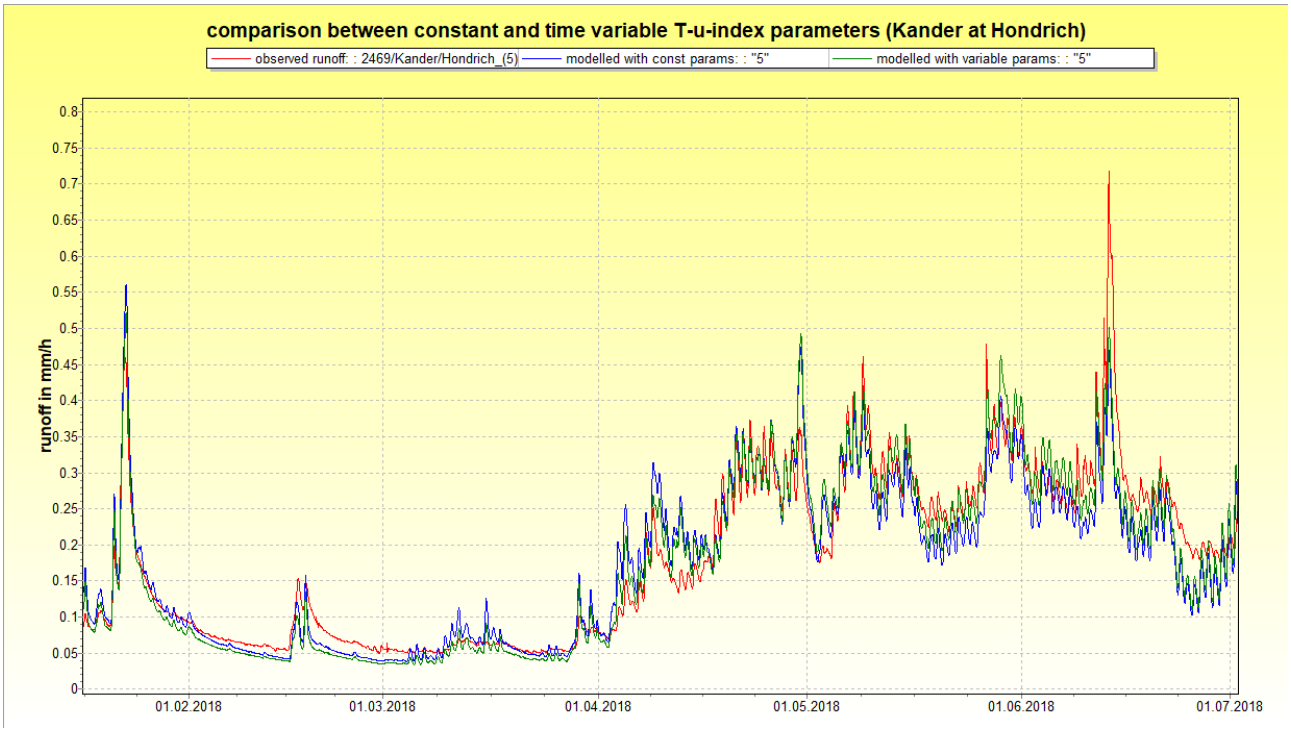


figure 2.9.8.1: comparison of modeled runoffs versus observed runoff (red) for blue=constant T-u-index parameters and green = temporally variable parameters

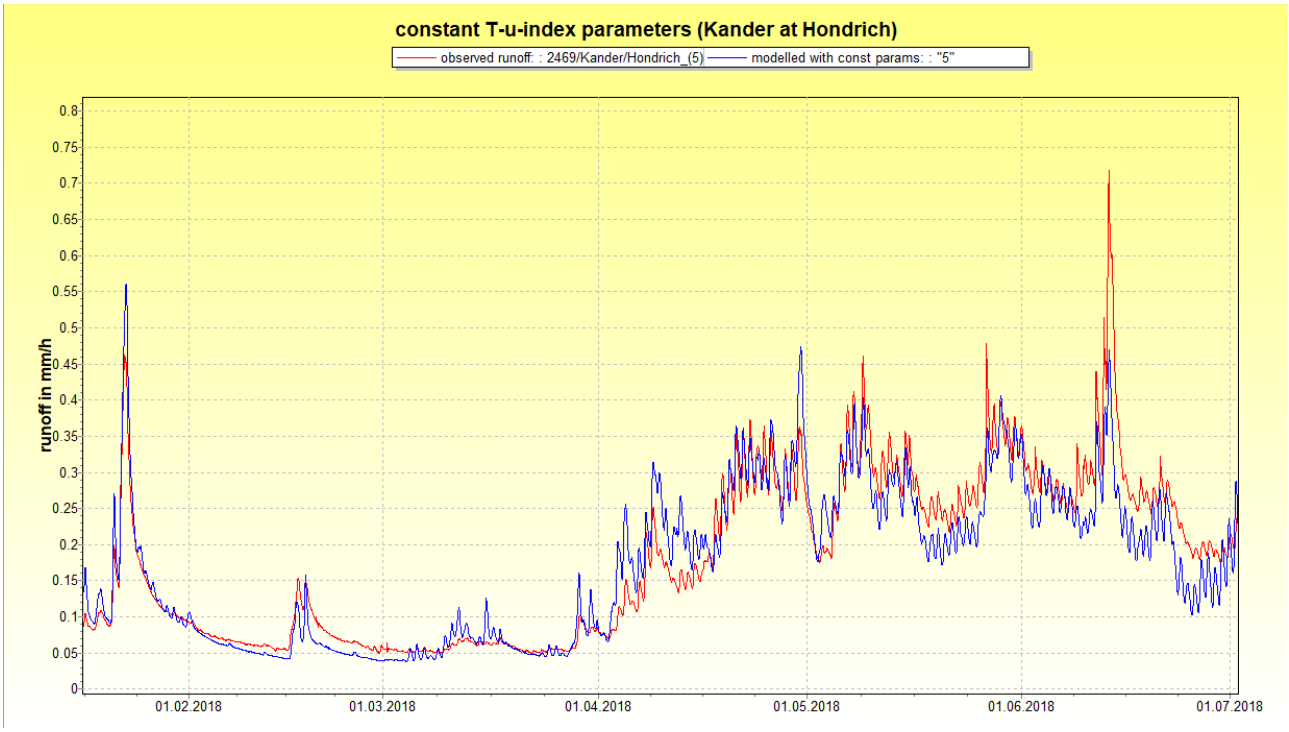


figure 2.9.8.2: Comparison of observed runoff (red) vs. modelled runoff (blue) for default static snow melt parameters

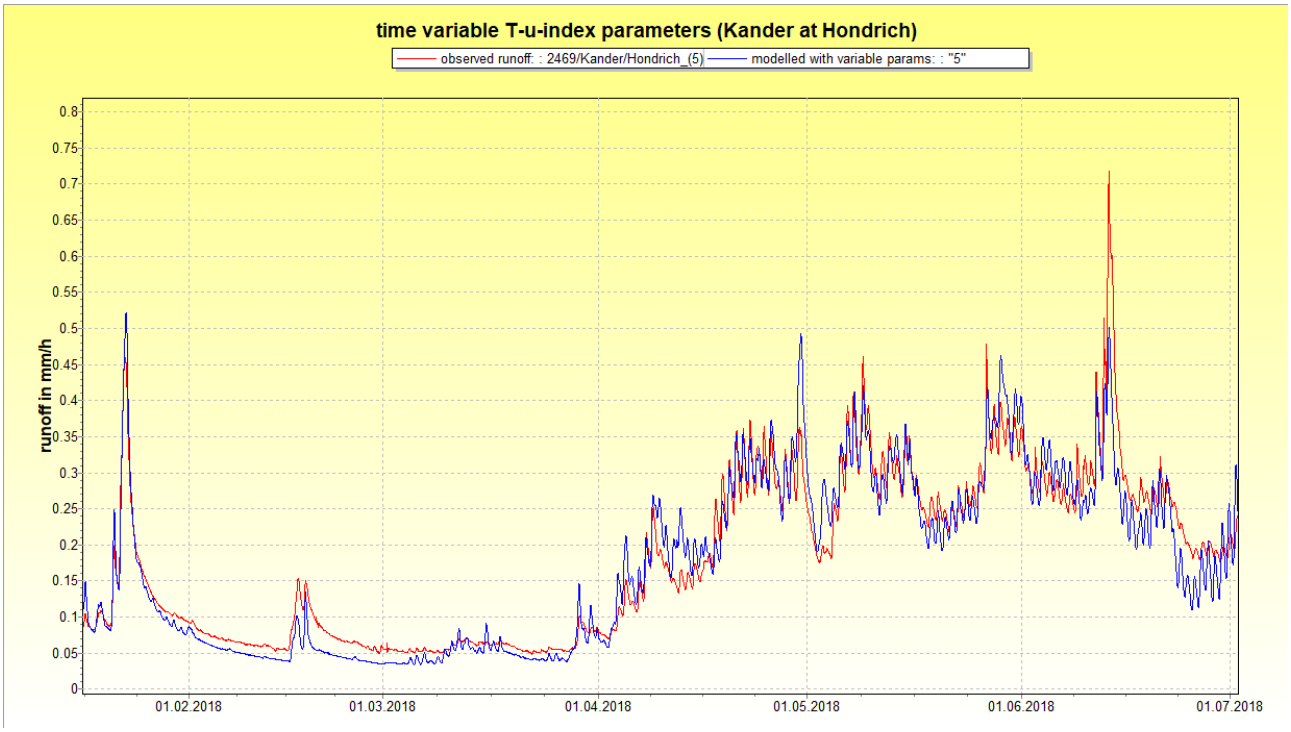


figure 2.9.8.3: Comparison of observed runoff (red) vs. modelled runoff (blue) for temporally variable snow melt parameters

2.10 Interception

2.10.1 Classic bucket approach for rain

A simple bucket approach is used with a capacity depending on the leaf area index, the vegetation coverage degree, and the maximum height of water at the leafs:

$$SI_{\max} = vcf \cdot LAI \cdot h_{SI} \qquad \textbf{(2.10.1)}$$

with
- SI_{max} maximum interception storage capacity [mm]
- vcf degree of vegetation covering (crop specific annual course) [m^2/m^2]
- LAI leaf area index (crop specific annual course) [m^2/m^2]
- h_{SI} maximum height of water at the leaf surfaces [mm]

In WaSiM, the interception is calculated after the snow model, so the snow melt is flowing into the interception storage in order to be able to store melt water (as well as rain water) at the soil surface in small ponds etc. This has to be done by configuring a leaf area index > 0 and a vegetation covered fraction of >0 for the snow covered vegetation (there is no interception on the bare soil, even if the old documentation described it, it was never implemented). During snow-free periods, the interception model models interception of water in the vegetation layer. There is no snow interception model yet implemented.

The extraction of water out of the interception storage by evaporation is assumed to be at potential rate. If there is a sufficient amount of water in the storage, the storage content is reduced by the potential evaporation and no evaporation will be taken from the soil. If the storage content is smaller than the potential evaporation rate, the remaining rate will be taken from the soil (if the soil is not too dry or too wet).

$$
\begin{aligned}
&EI = TP && (\text{for } SI \geq TP \text{ in mm}), \quad && TR = 0 \\
&EI = SI && (\text{for } SI < TP \text{ in mm}), \quad && TR = TP - SI \\
&SI_{new} = SI_{old} - EI
\end{aligned}
\qquad \textbf{(2.10.2)}
$$

with EI interception evaporation [mm]
TP potential transpiration [mm]
TR remaining potential transpiration from the vegetation, a reduction after eq. (2.7.1) or (2.7.2) dependent on soil moisture is still possible [mm]
SI content of the interception storage [mm]

Equation (2.10.2) describes the extraction of water from the interception storage for the traditional method (method code 1), where no differentiation exists between potential crop-transpiration and interception evapotranspiration.

However, if the newer method 2 is applied, then the real interception evaporation has to take into account not only the potential crop-transpiration but also the potential interception evaporation calculated by the same algorithms but with other resistances. Since only the maximum of both entities (and not their sum) can be evaporate, there has to be a balancing:

$$
\begin{aligned}
&EI = IP && \text{and} && TR = 0 && \text{for } SI \geq EIP \geq TP \ (\text{in mm}) \\
&EI = IP && \text{and} && TR = TP - EI && \text{for } SI \geq TP \geq EIP \ (\text{in mm}) \\
&EI = IP && \text{and} && TR = TP - EI && \text{for } TP \geq SI \geq EIP \ (\text{in mm}) \\
&EI = SI && \text{and} && TR = TP - SI && \text{for } SI \leq EIP \leq TP \ (\text{in mm}) \\
&EI = SI && \text{and} && TR = TP - SI && \text{for } SI \leq TP \leq EIP \ (\text{in mm}) \\
&EI = SI && \text{and} && TR = 0 && \text{for } TP \leq SI \leq EIP \ (\text{in mm}) \\
&SI_{new} = SI_{old} - EI && && && \text{for all cases above}
\end{aligned}
\qquad \textbf{(2.10.3)}
$$

with EI interception evaporation [mm]
EIP potential interception evaporation after e.g. (2.6.19 a) / (2.6.19 b) [mm]
TP potential transpiration [mm]
TR remaining potential transpiration from the vegetation, a reduction after eq. (2.7.1) or (2.7.2) dependent on soil moisture is still possible [mm]
SI content of the interception storage [mm]

The outflow of the interception model consists of two components:

1)the fraction of precipitation falling on the uncovered fraction of the cell $P \cdot (1\text{-}vfc)$

2)the outflow from the interception storage, if there is more precipitation than the interception storage can hold $(P\text{-}(SI_{max}\text{-}SI)) \cdot vcf$, if $P + SI > Si_{max}$, otherwise 0

$$
\begin{aligned}
&Q = P \cdot (1 - vcf) + (P - (SI_{max} - SI_{act})) \cdot vcf \quad \text{when} \quad P + SI_{act} > SI_{max} \\
&\rightarrow Q = P - vcf \cdot (SI_{max} - SI_{act}) \\
&Q = P \cdot (1 - vcf) \quad \text{when} \quad P + SI_{act} \leq SI_{max}
\end{aligned}
\qquad \textbf{(2.10.4)}
$$

with Q outflow from interception model [mm]
P precipitation or snow melt [mm]
SI_{max} maximum interception storage [mm]
SI_{act} actual interception storage after balancing the input with the storage capacity [mm]

The amount given by equation (2.10.3) or (2.10.4) is the input for the infiltration and/or soil model. Even if the interception storage is full, the extraction by evaporation is taken from the interception storage only after balancing input and output, thus regarding the fact, that usually (at least for shorter time steps) there is no evaporation at all during precipitation events.
If WaSiM runs with a multi-layer land use, the equations given above are applied on a per-layer

basis. The outflow of the uppermost layer is taken as input into the next lower layer. There, the same algorithms will be applied. Since the potential evapotranspiration rates were already calculated by taking into account the reduced radiation and cumulative LAI, the real evapotranspiration for the next lower layer may be added simply to the cumulative real evaporation from all layers above. However, it may be required to calibrate the model carefully for this multi-layer land use.

2.10.2 Control file settings for interception

The parameterization of the interception is only partly be done in the interception_model-section of the control file. The value of h_{SI} in (2.10.1) will be taken from the land use table and may be defined there with an annual course. However, if no value is given there or if the old land use table (up to WaSiM version 5) is used, the default value of h_{SI} is taken from the storage capacity parameter of the [interception_model] section of the control file.

```
[interception_model]
1                           # 0=ignore this module, 1 = run the module
$time                       # duration of a time step in minutes
1       # method: 1 = use ETP for calculating EI; 2 = use EIP for calculating EI (only effective for method 1 in
# evapotranspiration model -> for other methods, ETP = EIP)
$outpath//$throughfall      # result grid :  = outflow from the interception storage
$Writegrid                  # 0, if no grid-output is needed, else one of the codes described above
$outpath//qi__//$grid//.//$year $hour_mean # statistic file interception storage outflow
$outpath//$EIgrid           # Interzeption evaporation, grid
$Writegrid                  # 0, if no grid-output is needed, else one of the codes described above
$outpath//ei__//$grid//.//$year $hour_mean # zonal statistic
$outpath//$SIgrid           # storage content of the interception storage
$Writegrid                  # 0, if no grid-output is needed, else one of the codes described above
$outpath//si__//$grid//.//$year $hour_mean # zonal statistic For interception storage content
0.35                        # layer thickness of the waters on the leaves (multiplied with LAI -> storage capacity)
$readgrids                  # 1=read grids from disk, else generate internal
```

2.10.3 Advanced interception of snow, including a canopy energy balance model

The model components for snow interception and canopy energy balance were developed by Kristian Förster (snow interception; at Department of Geography, University of Innsbruck, Austria and later at the Institute of Hydrology and Water Resources Management, Leibniz University Hannover, Germany) and Matthias Kopp (canopy energy balance; Hydrology and River Basin Management, Department of Civil, Geo and Environmental Engineering, Technical University of Munich, Germany) in 2017 and 2018 and have been merged with the WaSiM main release branch by Jörg Schulla in early 2019. Detailed descriptions of the model and some examples are given in Förster (2018, https://doi.org/10.1080/02626667.2018.1518626) and Kopp (2017), see also WaSiM homepage for references. The following sub-chapters are mostly copied from these sources.

2.10.3.1 Snow canopy interception

Forest cover is a major cause of the spatial variability of snow accumulation and snowmelt, especially in subarctic environments and subalpine mountain regions with a high proportion of forest areas. Forests can thereby store considerable amounts of snow. E.g. boreal forest canopies can intercept up to 60% of cumulative seasonal snowfall. Published values for the maximum interception storage capacity of a conifer forest range from a few millimetres to 40 mm of snow water equivalent (SWE), depending on snow and climate conditions. Generally, trees have a higher interception capacity for snow than for rain Intercepted snow may be subject to high sublimation rates depending on atmospheric conditions. The amount of sublimation loss depends on the exposure time of snow in the canopy and can rise up to 30% of cumulated snowfall. The remaining 70% of snow in the canopy can be released from the branches as meltwater drip or mass unload, leading to highly non-uniform snow depth patterns underneath the canopy with generally reduced amounts of snow in the forest. Numerous studies have shown both delaying and accelerating effects of snowmelt have

been described. Decreased global radiation due to shading by trees and reduced wind speed inside the forest may delay snowmelt, whereas increased longwave emission by the trees accelerates snowmelt and becomes an important component of the surface energy balance. Which opposing effect dominates depends on several factors, such as canopy density, gap size and distribution, geographical location, aspect and meteorological conditions .

For the meteorological variables air temperature, relative humidity, wind speed and incoming shortwave radiation, a considerable heterogeneity can be observed beneath the trees. The simulation of melt rates and the timing of meltwater release from forested areas can hence be significantly improved by including local meteorological observations. Since these are mostly not available, transfer algorithms are used to calculate sub-canopy climatic conditions from open-field site measurements. A precise representation of sub-canopy climatic conditions is particularly important for the simulation of the snow surface energy balance at the forest ground.

The new model extension also includes a modification of the sub-canopy meteorological conditions since they differ significantly from those in the open. Beneath the trees, the shortwave radiation, precipitation and wind speed are reduced, longwave radiation and humidity are increased, and the diurnal course of air temperature is attenuated. As measurements of the meteorological variables are usually taken at observation sites in the open, the recordings have to be scaled to the conditions inside the canopy. The reduction formulas are applied for the interpolated meteorological fields of solar and thermal radiation, air temperature, humidity and wind speed. The only parameter required is the effective leaf area index, LAI_{eff} (-), e.g. from the literature, which is referred to herein as *LAI* for simplicity, i.e. including the area of the leaves or needles, the branches and the stem.

Beneath canopy climate

Solar radiation reaching the ground surface $Q_{sc\downarrow}$ (Wm^{-2}) is calculated as a fraction of top-of-canopy incoming solar radiation $Q_{s\downarrow}$ transmitted through the trees depending on *LAI*:

$$Q_{sc\downarrow} = Q_{s\downarrow} \times \exp(-0.71 \cdot LAI) \qquad \textbf{(2.10.5)}$$

where 0.71 is a dimensionless extinction coefficient. Longwave radiation reaching the ground $Q_{lc\downarrow}$ ($W \cdot m^{-2}$) consists of a fraction of the top-of-canopy incoming longwave radiation $Q_{l\downarrow}$, and a fraction of longwave radiation emitted by the trees:

$$Q_{lc\downarrow} = (1 - F_c) \cdot Q_{l\downarrow} + F_c \sigma T_c^4 \qquad \textbf{(2.10.6)}$$

where σ is the Stefan-Boltzmann constant, T_c (K) the canopy air temperature and F_c (-) the canopy density, which was empirically derived from a comprehensive forest structure dataset:

$$F_c = 0.55 + 0.29 \ln LAI \qquad \textbf{(2.10.7)}$$

For the calculation of T_c the dampening effect of the shading during the day and the emission of thermal radiation during the night are considered:

$$T_c = T_a - F_c \{T_a - [R_c (T_a - T_{mean}) + T_{mean} - \delta T]\} \qquad \textbf{(2.10. 8)}$$

where T_a (K) is the top-of-canopy air temperature, $R_c = 0.8$, T_{mean} (K) is the mean daily air temperature and δT is limited to the range $-2\ K \leq \delta T \leq +2\ K$:

$$\delta T = \frac{(T_{mean} - 273.16)}{3} \qquad \textbf{(2.10. 9)}$$

The increase of relative humidity RH_c (%) inside a canopy due to sublimation and evaporation of melting snow is modified:

$$RH_C = min\{RH(1+0.1F_c), 100\} \quad \textbf{(2.10. 10)}$$

For melt conditions, RH_c is set to saturation. Wind speed W_c (m s^{-2}) inside a canopy is calculated based on:

$$W_c = W_a \exp(-0.4 f_i) \quad \textbf{(2.10. 11)}$$

where f_i is the canopy flow index:

$$f_i = \beta \times LAI \quad \textbf{(2.10. 12)}$$

with β (= 0.9) being a dimensionless scaling factor.

Canopy snow processes

Snow that is intercepted on the leaves/needles, branches and stem of a tree can sublimate into the atmosphere, or – if the air temperature is above 0°C – meltwater drips and falls to the ground can occur. The unloading of snow by wind is not (yet) considered. Absorption of solar radiation SR_{abs} (W) by snow in the canopy is given by:

$$SR_{\text{abs}} = \pi r^2 (1-\alpha) Q_{s\downarrow} \quad \textbf{(2.10. 13)}$$

where the radius of a spherical ice particle r (m) is assumed to be 500 µm; and $Q_{s\downarrow}$ (W m^{-2}) is the top-of-canopy incoming solar radiation. For the intercepted snow particle albedo α, we assumed the same value as for simulated snow surface albedo in the open:

$$\alpha = \alpha_{min} + (\alpha_{t-1} - \alpha_{min}) \exp(-k \frac{1}{24}) \quad \textbf{(2.10. 1 4)}$$

where α_{min} is the minimum albedo of (old) snow; α_{t-1} is the albedo in the previous time step; and k is a recession factor depending on air temperature (which determines snow surface temperature). The factor 1/24 is required to scale the result to the hourly progression of the computations (and is internally set according to the actual time step). At every time step with a considerable snowfall (at least 0.5 mm h^{-1}), the snow albedo is reset to its maximum value α_{max}. For the calculation of the mass loss rate from the interception storage, the model requires the Reynolds, Nusselt and Sherwood numbers. The Reynolds number, Re, for 0.7 < Re < 10, is given by:

$$\text{Re} = \frac{2 r W_c}{\nu} \quad \textbf{(2.10. 1 5)}$$

where v is the kinematic viscosity of air (1.3 × 10^{-5}m^{2}s^{-1}). The Nusselt number, Nu, is given by:

$$\text{Nu} = 1.79 + 0.606 \sqrt{\text{Re}} \quad \textbf{(2.10. 1 6)}$$

In contrast to the original version, we calculated the saturation vapour pressure over ice e_s (Pa) using:

$$e_s = \begin{cases} 6.122 \cdot \exp\left(\frac{17.62(T-273.15)}{243.12+(T-273.15)}\right) & \text{if } T \geq 273.15 \\ 6.122 \cdot \exp\left(\frac{22.46(T-273.15)}{272.62+(T-273.15)}\right) & \text{if } T < 273.15 \end{cases} \quad \textbf{(2.10. 1 7)}$$

Absolute humidity at saturation ρ_v (kgm^{-3}) was calculated by:

$$\rho_v = 0.622 \frac{e_S}{R_d \cdot T} \quad \textbf{(2.10. 1 8)}$$

where the value of the gas constant for dry air, R_d, is 287 $JK^{-1}kg^{-1}$. The diffusivity of water vapour in the atmosphere D_v (m^2 s^{-1}) was computed after Thorpe and Mason (1966):

$$D_v = 2.06 \cdot 10^{-5} \left(\frac{T_c}{273} \right)^{1.75} \quad \textbf{(2.10. 1 9)}$$

For the calculation of the mass loss rate ($\partial m/\partial t$) both temperature and humidity are assumed to be constant with height:

$$\frac{\partial m}{\partial t} = \frac{2 \pi r \left(\frac{RH_C}{100} - 1 \right) - SR_{abs} \cdot \Omega}{l_s \Omega + \frac{1}{D_v \rho_v Sh}} \quad \textbf{(2.10. 20)}$$

where ls, the latent heat of sublimation, is 2.838 × 106 J kg^{-1}. In this computation, the Sherwood number, Sh, is set to Nu, and Ω is computed using:

$$\Omega = \frac{1}{\lambda_t T_c N_u} \cdot \left(\frac{l_s M_W}{RT_c} - 1 \right) \quad \textbf{(2.10. 21)}$$

where λ_t is the thermal conductivity of the atmosphere (0.024 J $m^{-1}s^{-1}K^{-1}$); M_w is the molecular weight of water (0.018 kg $mole^{-1}$); and R is the universal gas constant (8.313 J $mole^{-1}$ K^{-1}). The sublimation loss rate coefficient ψ_s (s^{-1}) can now be computed using:

$$\psi_s = \frac{\frac{\partial m_{sp}}{\partial t}}{m_{sp}} \quad \textbf{(2.10. 22)}$$

where m_{sp} (kg) is the particle mass (ψ_i is the ice density = 916.7 kg m^{-3}):

$$m_{sp} = \frac{4}{3} \pi \rho_i r^3 \quad \textbf{(2.10. 23)}$$

The maximum snow interception storage capacity I_{max} (mm) and the amount of solid precipitation in the current time step t, P (mm), were used to calculate the canopy-intercepted load I, given by:

$$I = I_{t-1} + 0.7 (I_{max} - I_{t-1}) \left(1 - \exp \left(\frac{-P}{I_{max}} \right) \right) \quad \textbf{(2.10. 24)}$$

Rain is assumed to fall through to the rain interception model (to account for interception in the understorey) and finally to the ground snow cover. The maximum interception storage capacity I_{max} is assumed to be 4.4 × LAI, and the sublimation loss rate Q_{cs} (mm) for the intercepted snow within the canopy is:

$$Q_{cs} = C_e I \psi_s \mathrm{d}t \quad \textbf{(2.10. 2 5)}$$

Sublimation thus only occurs at the surface of the intercepted snow. This is accounted for with the nondimensional canopy exposure coefficient C_e:

$$C_e = k_C \left(\frac{I}{I_{max}} \right)^{-0.4} \quad \textbf{(2.10. 2 6)}$$

The shape of the intercepted snow is described by the dimensionless coefficient k_C=0.01. For snow that is removed by snowmelt unload, it is assumed that initial melt at its surface is required. The rate of melt unload L_m ($kg \cdot m^{-2}$) is calculated using the temperature index melt model of Pellicciotti et al.

(2005), accounting for shortwave radiation and albedo:

$$L_m = \frac{C_t}{\Delta t}(T - 273.15) + \frac{C_a}{\Delta t}(1-\alpha) Q_{sc\downarrow} \Delta t \qquad \textbf{(2.10. 2 7)}$$

where C_t is the temperature factor (0.05 mm·h^{-1} ·K^{-1}), and C_α the albedo factor (0.0094 m^2 ·mm·Wh^{-1}). The unloading rate is limited to 5 kg·m^{-2} ·d^{-1} ·K^{-1}. The unloaded mass is added to the ground snow cover beneath the trees. In order to distinguish liquid from solid unload, an empirical computation
of the load of snow (solid fraction) falling to the ground is introduced:

$$L_{m,solid} = 2.3 \cdot L_m \qquad \textbf{(2.10. 2 8)}$$

In the extended WaSiM version, liquid outflow L_m is redirected to the rainwater interception model in order to account for rainwater interception in the understory. The solid part $L_{m,solid}$, however, is accumulated to the snow pack in the snow model.

The new model extension includes three parameters that have been estimated a priori without any further consideration in the subsequent calibration procedure. A threshold value has to be provided for *LAI* and the roughness length z_0, respectively. Here, LAI_{min} = 1.0 and $z_{0;min}$ = 0.1 m. Additionally, a scaling factor n_{LAI} is mandatory for increasing the internal *LAI* values used for evapotranspiration
calculation in WaSiM in order to consider the effective *LAI* for snow interception. The dense structure of needles and branches at the scale of a tree generally results in higher interception storage capacity for snow, exceeding typical values for rain. *LAI* values provided in the lookup tables in the WaSiM settings (originally used for rain interception and evapotranspiration) are hence corrected by a scaling factor of n_{LAI} = 1.2 in order to account for the effects described above. This value leads to effective *LAI* values comparable with those in the literature for similar forests.

2.10.3.2 Canopy energy balance

While the canopy snow interception model as described above provides not only an alternative algorithm for mass interception but also provides changed meteorologic input variables for the snow cover on ground, the main variable affecting the energy balance on the ground snow surface, the incoming long-wave radiation, is only empirically estimated, see equations (2.10.6) to (2.10. 9) . By computing a physically based energy balance for the canopy layer, this term can be estimated with a higher accuracy and thus replaces the incoming long-wave radiation as calculated by equation (2.10.6) . The following description is based on the technical report provided by Matthias Kopp, who implemented this algorithm in WaSiM. The actual implementation will probably change a bit, since the solution of the energy balance is an iterative process which not always converges, so the conditions for convergence are probably often better if e.g. the latent heat flux is neglected. Also, the heat input by precipitation, especially if it rains on a prevailing snow interception layer, is very

hard to estimate for longer time steps. That is the reason for some ongoing changes, so some of the following energy balance terms will probably be set to zero in a final version (however, version 10.04.01 is already stable, although it models sometimes unrealistic canopy temperatures).

The implemented canopy energy balance (CEB) is a so called single layer canopy energy balance model. This means that the canopy structure is simplified regarded as a single layer above the ground featuring certain physical properties. The layer is supposed to be in full physical contact with the overlaying air mass. Energy is exchanged in both directions via radiation and different forms of heat fluxes. The connection to the ground or respectively the snow cover is less physically based. The sub-canopy air conditions are computed within the new “Snow Canopy Model (SCN)” introduced by FÖRSTER ET AL. (2018, see chapter 2.10.3.1) using a conceptual approach. The

optical properties of the canopy layer alter the radiation balance for the underlying snow cover. Of special interest is the canopy temperature Tc, which is used as the variable during the iterative solving procedure for the canopy energy balance. The resulting canopy temperature determines the amount of emitted long-wave radiation from the canopy into the snow cover. This can lead to significant alterations in the snow cover energy balance and therefore to considerable changes in the snow ablation regime of the snow cover as will be shown below.

The energy balance equation for the canopy layer which is solved in the newly implemented "Canopy energy balance" model is:

$$0 = SW_{net} + LW_{net}[T_{can}] + H[T_{can}] + ¿[T_{can}] + A[T_{can}] + BM[T_{can}] \quad \textbf{(2.10. 2 9)}$$

with: SW_{net} net shortwave radiation [W/m^2]
LW_{net} net longwave radiation [W/m^2]
H sensible heat flux [W/m^2]
LE latent heat flux [W/m^2]
A advective energy flux by precipitation [W/m^2]
BM biomass heat storage [W/m^2]

Except for the net shortwave radiation every term is a function of the canopy temperature T_{can}. The following section describes the implemented formulations for the computation of the different heat fluxes of the canopy energy balance.

Shortwave Radiation

TACONET ET AL. (1986) give a formulation for the shortwave radiation balance of the canopy:

$$SW_{net} = SW_{in}(1-\alpha_{can})\sigma_f\left(1+\frac{\alpha_{surf}(1-\sigma_f)}{1-\sigma_f\,\alpha_{surf}\,\alpha_{can}}\right) \quad \textbf{(2.10. 30)}$$

with: SW_{net} net shortwave radiation of canopy layer [W/(m^2 ·K)]
SW_{in} incoming shortwave radiation (global Radiation) [W/(m^2 ·K)]
σ_f absorption factor [-]
α_{can} albedo value of the canopy[-]
α_{surf} albedo value of the ground surface [-]

The albedo value of the canopy is calculated in the SCN (surface canopy interception) module. The absorption factor σ_f describes the amount of absorbed radiation by the canopy given by a formulation of the Beer-Lambert law:

$$\sigma_f = 1 - e^{-k_{LAI}\,LAI} \quad \textbf{(2.10. 3 1)}$$

with: LAI leaf area index [m^2/m^2]
k_{LAI} extinction parameter (normally between 0.4 and 0.8) [-]

Longwave Radiation

Assuming an emissivity value of 1, according to TACONET ET AL. (1986) the net longwave radiation balance of the canopy layer can be calculated with:

$$LW_{net} = \sigma_f\left(LW_{in} + \sigma\,T_{surf}^4 - 2\,\sigma\,T_{can}^4\right) \quad \textbf{(2.10. 3 2)}$$

with: $LW_{net;can}$ net longwave radiation of canopy layer [W/(m^2 ·K)]
σ_f absorption factor [-]
LW_{in} incoming longwave radiation [W/(m^2 ·K)]
σ Stefan-Botlmann constant $5.67 \cdot 10^{-8}$ [W/(m2·K^{-4})]
T_{surf} ground surface Temperature [K]

T_{can} canopy layer temperature [K]

Sensible Heat

For the computation of the sensible heat flux between the top-of-canopy air mass and the canopy layer a bulk transfer formulation of the sensible heat flux is applied. In this approach the turbulent energy exchange processes caused by wind are described by the use of the aerodynamic resistance of the canopy r_a. The bulk transfer coefficient C_h takes atmospheric stability conditions into account.

$$H = \frac{C_h \rho_a c_p}{r_a}(T_a - T_{can}) \qquad \textbf{(2.10. 3 3)}$$

$$C_{hn} = k^2 \cdot (\ln(z_a / z_0))^{-2} \qquad \textbf{(2.10. 3 4)}$$

with:
H sensible heat flux [W/m^2]
σ_a air density [m/s^2]
c_p specific heat capacity of air [J/(kg·K)]
C_h bulk transfer coefficient [-]
r_a aerodynamic resistance [s/m]
T_a air temperature [K]
T_{can} canopy temperature [K]
C_{hn} bulk transfer coefficient under neutral stability conditions [-]
k Karman's constant [-]
z_a wind speed measurement height [m]
z_0 roughness length [m]

The stability conditions dependent on the air temperature and the temperature of the canopy. The conditions are stable, if the canopy is significantly cooler than the air. Hence during unstable conditions the canopy is warmer than overlaying air mass, causing much stronger turbulences. In case the temperatures of the canopy and the air are almost equal neutral stability conditions prevail. For the calculation of the correct value of the bulk transfer coefficient the atmospheric stability conditions have to be assessed first. Following the equations given by OKE (1988) the bulk transfer coefficient can be assessed using the dimensionless bulk Richardson number R_{ib}.

R_{ib} can be calculated with:

$$R_{ib} = \frac{2g}{T_a - T_{can}} \cdot \frac{z_0 (T_a - T_{can})}{u_a^2} \qquad \textbf{(2.10. 3 5)}$$

with:
R_{ib} Richardson number [-]
g gravitational acceleration [m/s^2]
T_a air temperature [K]
T_{can} canopy temperature [K]
z_0 roughness length [m]
u_a wind speed [m/s]

For stable conditions C_h is calculated with:

$$C_h / C_{hn} = (1 - 5R_{ib})^2 \qquad \textbf{(2.10. 3 6)}$$

and for unstable conditions C_h becomes:

$$C_h/C_{hn}=(1-16\,R_{ib})^{0.75} \qquad \textbf{(2.10. 3 7)}$$

Aerodynamic Resistance

The aerodynamic resistance r_a is calculated with:

$$r_a=\frac{\ln\left(\frac{z_m-d}{z_{0m}}\right)\cdot\ln\left(\frac{z_h-d}{z_{0h}}\right)}{k^2 u_a} \qquad \textbf{(2.10. 3 8)}$$

where u_a is the wind speed at height z_m and z_h is the height of the humidity measurement. According to BRUTSAERT (1982) d, z_{0m} and z_{0h} are:

$$\begin{aligned} d &= 0.67\cdot h \\ z_{0m} &= 0.123\cdot h \\ Z_{0h} &= 0.1\cdot z_{0m} \end{aligned} \qquad \textbf{(2.10. 3 9)}$$

Latent Heat

The amount of energy transferred by vaporization and condensation of water is modeled with:

$$LE=\frac{0.622\,L}{R_a T_{air}}\cdot\frac{1}{r_E}\cdot C_E(e_{sat}[T_{Can}]-e_{air}) \qquad \textbf{(2.10. 40)}$$

with:

L	heat of vaporization or condensation [J/(kg·K)]
R_a	specific gas constant for air [J/(kg·K)]
r_E	effective aerodynamic resistance [s/m]
$e_{sat}[T_{can}]$	saturation vapor pressure at canopy temperature [Pa]
e_{air}	air vapor pressure [Pa]

Effective aerodynamic Resistances

The effective aerodynamic resistance r_E is parametrized using the transpiration resistance r_{Etr} and the resistance against interception evaporation r_{Eint} weighted with the fraction of wet canopy:

$$\frac{1}{r_E}=\frac{1}{r_{Etr}}(1-f_{Wet})+\frac{1}{r_{Etr}}\cdot f_{wet} \qquad \textbf{(2.10. 41)}$$

The wet fraction of the canopy is determined by the fraction of the amount of intercepted water I and the interception capacity I_{max} assessed in the SCN module:

$$f_{wet}=(I/I_{max})^{2/3} \qquad \textbf{(2.10. 42)}$$

The values of r_{Etr} and r_{Eint} have to be set in the WaSiM landuse table.

Precipitation Heat

The last heat flux that is considered in the canopy energy balance is the advective energy flux by precipitation to the canopy A. Following the approach of WARSCHER (2014) for the computation of the precipitation heat for the energy balance of the snow cover the precipitation heat is:

$$A=\frac{c_w\cdot P_{rain}}{\Delta t}\cdot(T_{air}-T_{can})+c_{ice}\frac{P_{snow}}{\Delta t}(T_{air}-T_{can}) \qquad \textbf{(2.10. 43)}$$

c_w and c_{ice} are the specific heat capacities for water and ice respectively.

Biomass Heat Storage

The amount of energy stored within the biomass of the canopy layer between two time steps BM_{can} can be accounted for by calculating:

$$BM_{can} = HM_{can} \cdot \frac{T_{can,t} - T_{can,t-1}}{\Delta t} \qquad \textbf{(2 .10. 44)}$$

$T_{can,t}$ and $T_{can,t-1}$ are the canopy temperatures at the current and the preceding time step. The heat mass of the canopy layer HM_{can} is made up of the heat mass of leaves HM_{leaves} and the heat mass of the trunks HM_{trunk}:

$$HM_{leaves} = LAI \cdot d_{leaf} \cdot \rho_{biomass} \cdot C_{p,biomass} \qquad \textbf{(2 .10. 4 5)}$$

$$HM_{trunk} = 0.5 \cdot B \cdot z_{can} \cdot \rho_{biomass} \cdot C_{p,biomass} \qquad \textbf{(2 .10. 4 6)}$$

with:
- d_{leaf} eaf Thickness [m]
- $\rho_{biomass}$ density of biomass [kg/m^3]
- $C_{pbiomass}$ biomass specific heat mass [J/(kg·K)]
- B stand basal area [m^2 /m^2]
- z_{can} canopy height [m]

Iteration Procedure

The variable for solving the canopy energy balance is the canopy temperature T_{can}. Equation (2.10. 2 9) is solved by applying the very robust bisection method. The search for the solution of the CEB is restricted to the interval between 200 and 340 K. The tolerance ε for solving the balance equation is set to 10 $^{-8}$ K. In order to prevent non-physical solutions in case of bad input data the maximum temperature difference between the canopy and the air temperature is set to 30 K. For most situations the iteration converges after at most 10 iteration steps, often even less.

Simulation Results

In order to show the impact of the CEB on the snow storage development three different models are compared.

• 1: No SCN + no CEB (i.e. classic interception, method in control file =11)

• 2: SCN + no CEB (only snow canopy interception, without detailed energy balance, method = 21)

• 3: SCN + CEB (both new algorithms; method = 31)

All models use the energy balance method to model the snow cover (WARSCHER (2014)). Each model has been calibrated using a mixed objective function taking into account discharge data as well as SWE measurements from the catchment for the winter season 2016/17. Figure 2.10.3.1 displays the differences in the specific storage of snow water equivalent (SWE) within the Dreisaeulerbach catchment during the validation period in the winter season 2017/18. Neglecting the snow interception in the canopy (model 1) leads to an overestimation of stored SWE within the mostly forested catchment. Model 2 and 3 show a similar development of the snow storage during the winter season. Nevertheless during February and March differences of up to 30 mm of SWE can be observed. Figure 2.10.3.2 shows the resulting discharge behavior during the Winter season 2017/18.

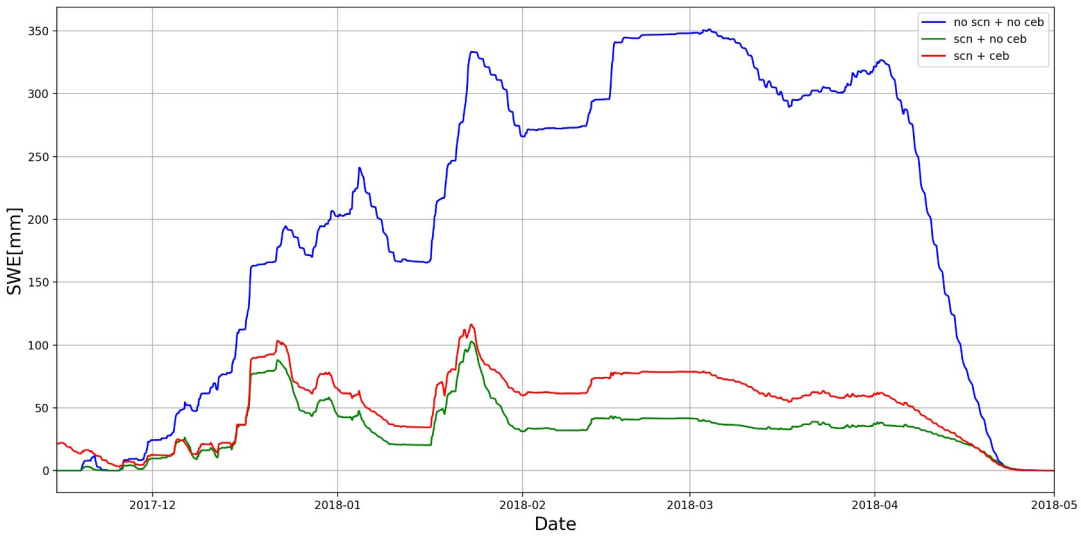


figure 2.10.3.1: Specific SWE stored in the catchment

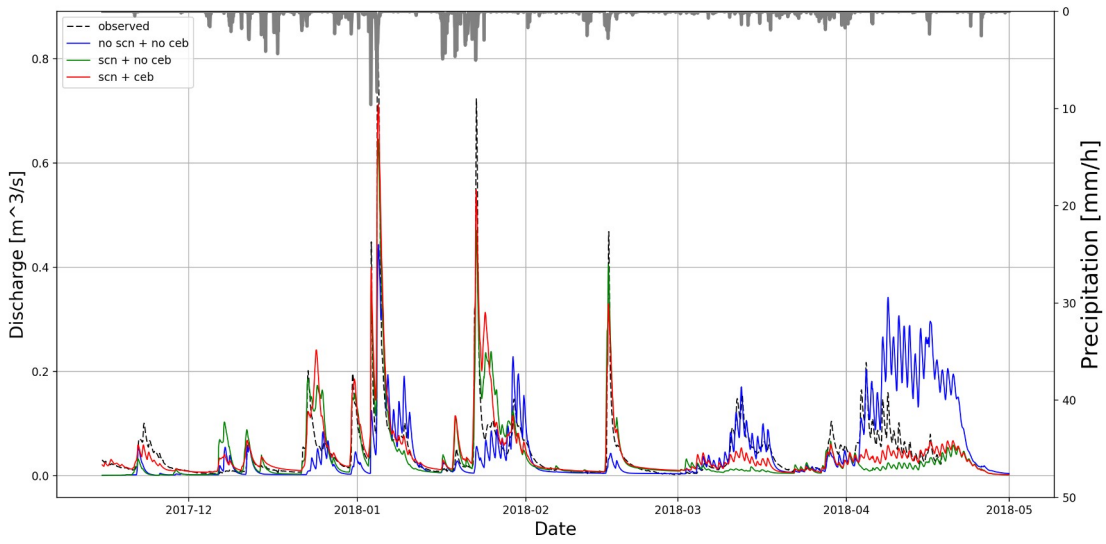


figure 2.10.3.2: Comparison of simulated and measured discharge data during validation run in winter 2017/18

It can easily be recognized that the parameterization of model 1 fails to reproduce the discharge behavior during the validation period. Taking into account the interception processes leads to a significant improvement of the model quality performance. The use of the CEB further improves the model quality. On the other hand, the iteration slows down the computation speed considerably.

2.10.4 Control file extensions for advanced interception

Any changes to the control file should still be backward compatible with older versions of WaSiM and also older control files should still run with new WaSiM versions, which is a kind of limited forward compatibility for the control file (i.e. old control files can be used in newer WaSiM versions but only without the new features). The key for this mechanism to work is the introduction

of new methods, having their parameters follow the known parameters. Here is an example of the new control file section for interception, including the old parameters for the conventional interception model (new part is in red, use methods 21, 22, 31 or 32 for any combination of the old and new methods):

```
 # methods for interception model, supported by version 10.04.00 and higher
# 1 = traditional/legacy: using ETP (no separate EIP), first snow model, then interception
# 2 = traditional/legacy: using separate EIP, first snow model, then interception
# 11 = using ETP (no separate EIP), first interception model, then snow model
# 12 = using separate EIP, first interception model, then snow model
# 21 = using ETP (no separate EIP), first interception model, then snow model, + Snow canopy interception SCN
# 22 = using separate EIP, first interception model, then snow model, + Snow canopy interception SCN
# 31 = using ETP (no separate EIP), first interception model, then snow model, + Snow canopy interception SCN +
Canopy energy balance CEB
# 32 = using separate EIP, first interception model, then snow model, + Snow canopy interception SCN + Canopy
energy balance CEB
#
[interception_model]
1                           # 0=ignore this module, 1 = run the module
$time                       # duration of a time step in minutes
31      # method: 1 = use ETP for calculating EI; 2 = use EIP for calculating EI (only effective for method 1 in
evapotranspiration model -> for other methods, ETP = EIP), 11 = like one but before snow model, 12=like 2 but
before snow model
$outpath//$throughfall        # result grid :  = outflow from the interception storage
1//$Writegrid                   # 0, if no grid-output is needed, else one of the codes described above
$outpath//qi__//$grid//.//$code//$year $hour_mean # statistic file interception storage outflow
$outpath//$EIgrid             # Interzeption evaporation, grid
$Writegrid                # 0, if no grid-output is needed, else one of the codes described above
$outpath//ei__//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//$SIgrid             # storage content of the interception storage
$Writegrid                    # 0, if no grid-output is needed, else one of the codes described above
$outpath//si__//$grid//.//$code//$year $hour_mean # zonal statistic For interception storage content
0.35                          # layer thickness of the waters on the leaves (multiplied with LAI -> storage
capacity)
$readgrids                    # 1=read grids from disk, else generate internal
# new parameters for snow interception model (implemented by Kristian Förster
# method must be 21, 22, 31 or 32 (where 31 and 32 will additionally switch on canopy
# energy balance, see above
$outpath//SCNTEMP//$grid//.//$suffix            # Sub-canopy temperature, grid
$Writegrid #_every                  # 0, if no grid-output is needed, else one of the codes described above
$outpath//SCNTEMP_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNRH//$grid//.//$suffix              # Sub-canopy relative humidity, grid
$Writegrid #_every                  # 0, if no grid-output is needed, else one of the codes described above
$outpath//SCNRH//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNRAD//$grid//.//$suffix             # Sub-canopy shortwave radiation, grid
$Writegrid #_every                # 0, if no grid-output is needed, else one of the codes described above
$outpath//SCNRAD_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNWIND//$grid//.//$suffix            # Sub-canopy temperature, grid
$Writegrid #_every                  # 0, if no grid-output is needed, else one of the codes described above
$outpath//SCNWIND_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNLOAD//$grid//.//$suffix            # Snow interception load, grid
$Writegrid #_every                # 0, if no grid-output is needed, else one of the codes described above
$outpath//SCNLOAD_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNAGE//$grid//.//$suffix             # Age of intercepted snow, grid
$Writegrid #_every                  # 0, if no grid-output is needed, else one of the codes described above
$outpath//SCNAGE_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNALB//$grid//.//$suffix             # Albedo of intercepted snow, grid
$Writegrid #_every                  # 0, if no grid-output is needed, else one of the codes described above
$outpath//SCNALB_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNSUBL//$grid//.//$suffix            # Sublimation of intercepted snow, grid
$Writegrid #_every                  # 0, if no grid-output is needed, else one of the codes described above
$outpath//SCNSUBL_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNUNLOAD//$grid//.//$suffix          # unload (incl. melt) of intercepted snow, grid
$Writegrid #_every                  # 0, if no grid-output is needed, else one of the codes described above
$outpath//SCNUNLOAD_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNLIQOUT//$grid//.//$suffix          # liquid water output of the snow canopy model, grid
$Writegrid #_every                  # 0, if no grid-output is needed, else one of the codes described above
$outpath//SCNLIQOUT_//$grid//.//$code//$year $hour_mean # zonal statistic
0.4         # minimum LAI value above which the snow canopy model will be started
0.16        # minimum Z0 value above which the snow canopy model will be started
1.52            # LAI scaling factor for the snow canopy model --> CALIBRATION!!
# new algorithms for interception canopy energy balance, implemented by Matthias Kopp, methods must be 31 or 32,
this includes snow canopy interception
$outpath//CEB_CANENBAL//$grid//.//$suffix
$Writegrid #_every
$outpath//CEB_CANENBAL//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//CEB_CANTEMP//$grid//.//$suffix
$Writegrid #_every                  #2
$outpath//CEB_CANTEMP//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//CEB_CANTEMP_OLD//$grid//.//$suffix
$Writegrid #_every
$outpath//CEB_CANTEMP_OLD//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//CEB_DUMMY//$grid//.//$suffix
$Writegrid #_every
$outpath//CEB_DUMMY//$grid//.//$code//$year $hour_mean # zonal statistic
```

2.11 Infiltration

The infiltration model is an integrated part of the soil model, but implemented in the Topmodel-version only. It uses an approach after PESCHKE (1977, 1987) which is based on the approach of GREEN and AMPT (1911). The soil is assumed to be homogeneous and not layered. matrix flow is assumed to dominate, the wetting front is approximated as a step function. The precipitation intensity is assumed to be constant over the entire time step. The approach consist of two phases. Within the first phase the time of saturation is calculated (if any saturation will occur within the time step at all). Within the second phase, the cumulative infiltration until the end of the time step is calculated. If the soil surface is saturated at the end of the time step, in the next time step only the cumulative infiltration will be calculated (if the precipitation intensity is sufficiently high) without calculating the saturation time. The exceeding amount (the not infiltrated water) is surface runoff. Using a parameter x_f the amount of re-infiltrating water can be controlled (0… 1). This may be important to consider inhomogeneities of the soil properties if using larger grid cells (e.g. for grid cell sizes of some hundreds of meters).

If $PI > K_s$ (otherwise there will be no saturation at all), the saturation time t_s is calculated by:

$$t_S = \frac{l_S \cdot n_a}{PI} = \frac{\frac{\Psi_f}{PI/K_S - 1}}{PI} \qquad \textbf{(2.11.1)}$$

with
- t_S saturation time from the beginning of the time step [h]
- l_S saturation depth [mm]
- n_a fillable porosity ($n_a = \Theta_S - \Theta$) [-]
- ψ_f suction at the wetting front ($\approx 1000\, n_a$) [mm]
- PI precipitation intensity [mm/h]
- K_S saturated hydraulic conductivity [mm/h]

The infiltrated amount of water up to this time F_S is given by:

$$F_S = l_S \cdot n_a = t_S \cdot PI \qquad \textbf{(2.11.2)}$$

Finally, the cumulated amount of infiltration after saturation until the end of the time step is calculated after PESCHKE (in DYCK, PESCHKE [1989]) by:

$$F = \frac{A}{2} + \left[\frac{A^2}{4} + AB + F_S^2 \right]^{1/2} \qquad \textbf{(2.11.3)}$$

with $A = K_S(t - t_s)$

$B = F_s + 2 \cdot n_a \cdot \psi_f$

The exceeding amount of precipitation $PI \cdot \Delta t - F - F_s$ is surface runoff $Q_{D,I}$.

2.11.1 Control file settings for the infiltration model

```
[infiltration_model]
0                           # 0=ignore this module, 1 = run the module
$time                       # duration of a time step in minutes
$outpath//$INFEXgrid        # grid with infiltration excess in mm (surface runoff)
$Writegrid                #  for surface discharge (fraction 1)
$outpath//infx//$grid//.//$year $hour_mean # statistic file for the infiltration excess
$outpath//$SATTgrid         # grid with code 1=saturation at interval start, 0 =no saturation.
$Writegrid                  # Writegrid for saturation code grids
0.1                         # fraction of reinfitrating water (of the infiltration excess)
```

2.12 Soil model for the WaSiM version using the Topmodel-approach

2.12.1 Introduction

The modeling of the soil water balance and of runoff generation is done using a modified variable

saturated area approach after BEVEN and KIRKBY (1979) extended by capillary rise and interflow. The calculation is done separately for each of the grid cells opposed to the modeling of classes of similar indices like in the original TOPMODEL. Basic data for the model is a spatial distribution of the topographic index c_s (or rather the soil-topographic index, when incorporating the transmissivity as we do it here). Using this index, the potential extent of saturation areas can be estimated depending on the mean saturation deficit within the basin:

$$c_s = \ln \frac{a_t}{T_0 \tan \beta_t} \qquad \textbf{(2.12.1)}$$

with c_s topographic index [-]
a_t specific catchment area per unit length of a grid cell;
this is the area draining through one meter of the edge of a grid cell [$m^2 \cdot m^{-1}$]
T_0 saturated local hydraulic transmissivity ($T_0 = \int K_s dh$) [$m^2 \cdot s^{-1}$]
β_t slope angle [$m \cdot m^{-1}$]

It is presumed, that (i) the groundwater table is parallel to the topographic slope, that (ii) the dynamic of the saturated zone can be approximated by subsequent quasi stationary states, and that (iii) the local hydraulic transmissivity T_h is a exponential function of the saturation deficit S:

$$T_h = T_0 e^{-S/m} \qquad \textbf{(2.12.2)}$$

The spatial distribution of saturated areas or, more general, the distribution of the saturation deficit can be found by:

$$S_i = S_m - m\left(\ln \frac{a_t}{T_0 \cdot \tan \beta_t} - \gamma\right) \qquad \textbf{(2.12.3)}$$

with S_m mean saturation deficit of the basin (arithmetic average of all S_i) [mm]
m model parameter [mm]
γ mean topographic index of the (sub-)catchment

At places where S_i is negative or zero, any further liquid precipitation (interception throughfall or snow melt) will immediately generate surface runoff. The mean saturation deficit is newly calculated in each time step as average value of all local saturation deficits from the previous time step and from the balance of all inflows into an outflows out of the (sub-)basin. After estimating all runoff components for the entire sub-basin, the new mean saturation deficit is calculated by:

$$S_{m,i} = S_{m,i-1} + Q_B + Q_r - Q_{SUZ} \qquad \textbf{(2.12.4)}$$

with $S_{m,i}$ spatially averaged Saturation deficit in the actual time step [mm]
$S_{m,i-1}$ spatially averaged Saturation deficit in the previous time step [mm]
Q_B base flow in the actual time step (is always a mean value for the (sub-)catchment [mm]
Q_r capillary rise as a mean value for the (sub-)catchment in the actual time step [mm]
Q_{SUZ} groundwater recharge from the unsaturated zone as a mean value for the (sub-)catchment [mm]

2.12.2 The course of water within the soil model

surface runoff from snow melt

If there is a sufficient snow cover on the ground (>10 mm water equivalent), a specified fraction of snow melt is taken as surface runoff directly from this melt (and is not routed to the infiltration module):

$$Q_{D,S} = Q_{\text{snow,out}} \cdot QD_{\text{snow}} \quad \textbf{(2.12.5)}$$

with $Q_{D,S}$ surface runoff (fraction from snow melt) [mm]
$Q_{snow,out}$ snow melt [mm]
QD_{Snow} factor defining the fraction of surface runoff on the snow melt [-]

partitioning of the water reaching the soil

Depending on the precipitation intensity, a greater or lesser fraction of precipitation can flow directly into deeper soil regions without going into the root zone storage (from where evaporation could remove the water again). The fraction is specified by a threshold value for the precipitation intensity:

$$P_{\text{grenz}} = P_{\text{grenz,1h}} \cdot \Delta t^{a_p} \quad \textbf{(2.12.6)}$$

with P_{grenz} threshold precipitation intensity, if $P > P_{grenz}$, all exceeding precipitation is routed into the deep soil by macro pores [mm/Δ T]
$P_{grenz,1h}$ threshold precipitation intensity for time step Δ t = 1h [mm/h]
Δ t time step [h]
a_p empirical value, describing the decrease of the variance of the precipitation intensity with larger time steps (≈ 0.6, static value in the model source code) [-]

All precipitation below the threshold value P_{grenz} flows into the root zone storage, which is also called soil storage, from where it can be withdrawn by evaporation after equation. The exceeding fraction is flowing into deeper soil regions immediately:

$$q_v = K_{\text{korr}} \cdot k_f \cdot e^{-S_i/m} \quad \textbf{(2.12.7)}$$

with q_v vertical flow rate (percolation) [mm]
K_{korr} scaling parameter for considering unsaturated soils as well as preferred flow paths [-]
S_i local saturation deficit [mm]
m recession parameter (see generation of baseflow) [mm]
k_f saturated hydraulic conductivity [mm/h]

Important: The parameter K_{korr} is used in the original-TOPMODEL as a parameter for considering unsaturated conditions. In WaSiM, this parameter is used also for considering preferential flow paths. Thus, this parameter can be within a wide range of values. K_{korr} should therefore be calibrated. For soils with much macropores, it should be much greater than for compact soils, otherwise the water will not reach the saturated zone fast enough.

also Important: If starting the model calibration with a combination of a much too small K_{korr} and/or with a too large initial saturation deficit S_m, this will lead to a kind of hydraulic separation as a result of the exponential function in equation (2.12.7) . The result for q_v will be extremely small and almost no water is able to percolate and all precipitation will leave the cell as evaporation or as surface runoff. It is therefore recommended to set K_{korr} at the beginning to a large number (e.g. 1000) and to set S_m to a rather small number (≥0)

generation of interflow and surface runoff from saturated areas

Interflow may be generated between soil layers of different hydraulic conductivities or porosities. There must be a sufficient slope, otherwise no interflow but water logging will be the result.

The WaSiM version with the TOPMODEL-approach considers interflow using a conceptual approach (because there is no soil-stratification information). Like the surface runoff, the interflow storage is filled depending on the local saturation deficit:

$$Q_{SH,\text{in}} = (S - SUZ) - SH_{\max} \qquad \textbf{(2.12.8)}$$

with $Q_{SH,in}$ inflow into the interflow storage (for each grid cell one storage) [mm]
S actual local saturation deficit [mm]
SUZ content of the **s**torage of the **u**nsaturated **z**one (pore volume, which is not accessible by plant roots down to the saturated zone) [mm]
SH_{max} maximum content of the interflow storage [mm], this is a model parameter

Surface runoff $Q_{D,sat}$ is generated only if the interflow storage is completely filled (what means also, that the unsaturated zone is filled, too, and hence there is no unsaturated zone at all). The original TOPMODEL with only one fast runoff component is get if $SH = 0$ mm. The total surface runoff is the sum of all possible three components: $Q_{D,S}$ from snow melt, $Q_{D,I}$ from infiltration excess and $Q_{D,sat}$ from saturated areas.

capillary rise from the saturated zone into the soil water storage and into the interflow storage

In order to allow evaporation with potential rates at places with high groundwater, the deficit in the soil water storage (root zone storage) caused by evaporation can be replaced by water from the saturated zone and/or from the interflow storage. The first re-flow is taken from the interflow storage proportional to its filling (Q_r). Next, the capillary rise will take water from the saturated zone until the evaporation losses are refilled (Q_{kap}). A hydraulic contact between the root zone and the saturated zone is required for this flow to take place. Two assumptions are made:

1. there is no capillary fringe above the groundwater – the decision, if capillary rise will take place depends only on the saturation deficit compared to the root zone capacity.
2. the extraction of water by evaporation is done using an unique intensity for the entire soil profile, thus avoiding a subdivision of the soil into layers.

A parameter r_k is used to quantify the re-flow into the soil storage (to scale the saturation deficit to the root zone depth). This is required due to the fact that the modeled groundwater table (given by the saturation deficit) does not fit well the real groundwater table as observed in the catchment:

$$Q_r = (ETR - Q_{kap}) \cdot \frac{SH}{SH_{max}} \cdot r_k \qquad (0 \le SH \le SH_{max}) \qquad \textbf{(2.12.9)}$$

$$Q_{kap} = \left(1 - \frac{S}{r_k \cdot n_e \cdot z_W}\right) \cdot ETR \qquad (0 \le S \le r \cdot n_e \cdot z_W) \qquad \textbf{(2.12.10)}$$

with Q_r amount of water flowing back from the interflow storage into the soil storage [mm]
Q_{kap} capillary rise from groundwater [mm]
ETR evaporation withdrawal from the soil [mm]
SH content of the interflow storage [mm]
SH_{max} maximum content of the interflow storage [mm]
S local saturation deficit [mm]
r_k scaling parameter (there will be no capillary rise for cells with $S > r_k \cdot n_e \cdot z_w$) [-]
n_e drainable porosity [-]
z_w root depth [mm]

calculating baseflow

The baseflow is calculated for each subbasin as a whole:

$$Q_B = e^{-\gamma + \ln T_{korr}} \cdot e^{-S_m/m} = T_{korr} \cdot e^{-\gamma} \cdot e^{-S_m/m} \qquad \textbf{(2.12.11)}$$

with Q_B baseflow [mm/time step]
γ mean value of the topographic index [-]
S_m mean saturation deficit for a (sub-)basin [mm]
m model parameter [mm]
T_{korr} scaling factor for transmissivities as well as for scale dependent shifts in the distribution function of the topographic index c [-]

For estimating the parameter m it is recommended to fit some appropriate recession periods in an inverted graph (in h/mm or days/mm) of the observed discharge by a linear regression, see equation (2.12.12) . The slope of this inverted recessions give the parameter m. **Important:** If modeling interflow, the parameter m usually has to be much larger than without using interflow (often the factor 2 or 3 has to be applied), because then the fast runoff components are handled by the interflow storage and m is responsible for the slower components only.

$$\frac{t}{m}+\frac{1}{Q_0}=\frac{1}{Q_B(t)} \qquad \textbf{(2.12.12)}$$

flow concentration within the sub-basin

The base flow is generated for an entire sub-basin as average value. Interflow is generated for each grid cell separately and then averaged over space. The third component, the surface runoff, is routed to the sub-basin outlet using a subdivision of the basin into flow time zones. Flow time zones are zones of equal flow times for surface runoff to reach the sub-basin outlet, where all grid cells belonging to the same zone have the same flow time (rounded up to the next integer number of time steps). The fraction of each of these zones on the total area, ordered by the zone number, can be seen as a system response. At the beginning of a new time step, the surface runoff of each flow time zone is given to the next lower flow time zone, only the content of the lowest zone is taken as effective surface runoff for the actual time step and added to base flow and interflow to get the total runoff. For considering retention, a single linear storage approach is applied to the surface runoff in the last flow time zone, using a storage constant k_D:

$$Q_{D_i}=Q_{D_{i-1}}\cdot e^{-\Delta t/k_D}+\hat{Q}_D\cdot\left(1-e^{-\Delta t/k_D}\right) \qquad \textbf{(2.12.13)}$$

with $Q_{D,i}$ transformed (by a single linear storage) surface runoff in the time step i [mm]
$Q_{D,i-1}$ transformed (by a single linear storage) surface runoff in the time step i-1 [mm]
$\hat{Q}_D$ surface runoff in the time step i within the lowest flow time zone [mm]
Δt time step [h]
k_D recession constant for surface runoff single linear storage [h]

At the begin of the next time step, the uppermost flow time zone is empty, whereas in the other flow time zones is already the runoff amount coming from the upper flow time zones. If there is newly generated surface runoff, it will be added to the amount which is already contained in the flow time zones.

2.12.3 Control file setting for the soil model (Topmodel version)

```
# this paragraph is not needed for WaSiM-uzr but for the WaSiM-version with the variable saturated
area approach (after Topmodel)
[soil_model]
1                       # 0=ignore this module, 1 = run the module
$time                   # duration of a time step in minutes
1                       # method, 1 = nortmal base flow, 2 = with slow baseflow (not recommended)
$outpath//$sat_def_grid  # (new) saturation deficite-grid (in mm)
$Writegrid              # writegrid for this grid
$outpath//$SUZgrid      # (new) storage grid for unsat. zone
```

```
$Writegrid               # writegrid for this grid
$outpath//$SIFgrid       # (new) storage grid for interflow storage
$Writegrid               # writegrid for this grid
$outpath//$SBiagrid      # (new) grid for soil moisture in the inaktive soil storage
$Writegrid               # Writegrid for inaktive soil moisture
$outpath//$fcia_grid     # (new) grid for plant available field capacity in the  inaktiven soil
storage
$Writegrid               # writegrid for this grid
$outpath//$SSPgrid       # (new) grid for the relative fraction of the soil storages, which is in
contact with ground water
$Writegrid               # writegrid for this grid
$outpath//$QDgrid        # (new) grid for surface runoff
$Writegrid            # writegrid for this grid
$outpath//$QIgrid        # (new) grid for Interflow
$Writegrid            # writegrid for this grid
$outpath//$Peakgrid      # (new) grid for Peakflow (maximum peakflow for the entire model time)
$outpath//qdir.sta $hour_mean # statistic of surface discharge
$outpath//qifl.sta $hour_mean # statistic of interflow
$outpath//qbas.sta $hour_mean # statistic of base flow
$outpath//qbav.sta $hour_mean # statistic of slow base flow
$outpath//qges.sta $hour_mean # statistic of total discharge
$outpath//sb.sta $hour_mean    # soil storage in mm per zone
$outpath//suz.sta $hour_mean   # drainage storage in mm per zone
$outpath//sifl.sta $hour_mean # interflow storage in mm per zone
$outpath//sd.sta $hour_mean # saturation deficite per zone in mm
   1       2       3      # Codes der Teilgebiete im Zonengrid
   0.025   0.045   0.042  # recession parameter m for Topmodel approach in m
   5.0     5.0     5.0  # correction factvor for transimissivities
   2.0     2.0     2.0  # correction factor for Ks (vertical)
   6.0     7.0     7.0  # SLR storage coefficient direct runoff in h
  25.0    25.0    25.0  # threshold of saturation deficit for interflow generation
  80.0    80.0    80.0  # SLR storage coefficient interfow in h
3600.0  3600.0  3600.0  # SLR storage coefficient slow base flow in h
   0.03    0.03    0.03 # maximum deep percolation (for slow base flow) at saturation in mm/h
   0.01    0.01    0.01 # initial value slow base flow QBB
   0.0     0.0     0.0  # initial value SUZ-storage in n*nFK (nFK = useable field capacity in mm)
   0.75    0.75    0.75 # useable field capacity in n*nFK,
   3.0     3.0     4.0  # threshold for macro pore runoff
   0.9     0.9     0.9  # reduction factor for re-loading the soil storage from interflow and
#groundwater storages (compensating evaporation losses at high groundwater or filled interflow
#storage)
   0.1     0.1     0.1  # fraction of snow melt which is not infiltrating but part of direct runoff
$readgrids          # 1=read grids from disk, else generate internal
$outpath//storage_topmodel.ftz    # if readgrids = 1, then this file contains the contents of the
# flow travel time zones for interflow and surface flow and for the tracers
```

2.12.4 Soil stability index (WaSiM-Topmodel-specific)

As a specific extension, (only) the Topmodel-version of WaSiM allows the computation of a "soil stability index". The implementation follows the documentation of SINMAP in PACK et al. (1998), which should be studied first to get detailed information on the soil stability index model.

For the soil stability index (SSI) to be calculated, a series of parameter grids has to be provided:

- root cohesion C_r [N/m^2]
- soil cohesion C_s [N/m^2]
- wet soil density ρ_s [kg/m^3]
- internal friction angle of the soil φ [-]
- soil depth D [m]
- local slope θ [-]

Using the Topmodel equations (especially 2.12.3), the water table depth D_w can be esitimated. All of the above entities will be used by the following equation to compute the Soil Stability Index (SSI):

$$SSI = \frac{C + \cos\theta \cdot (1 - w \cdot r) \cdot \tan\phi}{\sin\theta} \qquad \textbf{(2.12.14)}$$

where $w = D_w/D$ relative wetness $(0 \le D_w \le 1)$ [-] (

$r = \rho_w / \rho_s$ ratio of density of water to density of wet soil [-]

$C = (C_r + C_s)/(h \rho_s g)$ combined cohesion, made dimensionless by dividing by the static pressure of the soil (perpendicular to the slope)

This value of SSI is available in a standard grid, which can be written as desired at every time step or every day etc.

Extension of the control file when using soil stability index:

```
[standard_grids]
…
$inpath//$slope_grid    slope_angle      1 # slope angle (0..90°) -> used for soil stability index
$inpath//$RootCohesion  root_cohesion    1 # root cohesion for soil stability index [N/m^2]
$inpath//$SoilCohesion  soil_cohesion    1 # soil cohesion for soil stability index [N/m^2]
$inpath//$WetSoilDensity wet_soil_density 1 # wet soil density for soil stability index [kg/m^3]
$inpath//$InternalFrictionAngle internal_friction_angle 1 # soil internal friction angle [rad]
$inpath//$SoilDepth     soil_depth       1 # soil depth as used for soil stability index [m]

[variable_grids]
$outpath//$SoilStabilityIndex soil_stability_index 1  0 # soil stability index: default = 0?
$Writegrid                                              # Writegrid for this grid
$readgrids
```

2.13 Silting up model and Surface routing-module

Since version 8.01.02 WaSiM includes two new components: a surface routing component and a silting up (or coating/crusting) component. The following chapters will describe the functional design and the theoretical basis as well as the additionally required input data and parameters and changes to the control file. Since both modules share some data entities (like the surface runoff grid), they are described here in combination. However, it is well possible to run each of the modules without the other one. *Since version 10.00.00, a new surface routing algorithm that creates much smoother surfaces in much less (iteration) time is implemented, see chapter 2.13.7 ,*

2.13.1 Key features of both modules

2.13.1.1 Silting-up Module

- infiltration into soil is controlled by an exponential recession function
- main parameters for infiltration rate are
  - the cumulated kinetic energy of a rain event,
  - the silting-up disposition which depends on soil type and time since last tillage
  - and the end infiltration rate depending on the soil type (fractions of silt, sand and clay)
- after a rain pause which is long enough for recreation of the soil, the cumulated kinetic energy is set to 0 again.
- Actual infiltration rate using the above mentioned parameters is compared to the actual precipitation rate → excessive water will be branched off as a surface runoff component
- remaining water is transferred to the unsaturated zone model as input
- there, infiltration excess or even exfiltration from groundwater will be added to the surface discharge from the silting up module.
- No parameter change for the soil model necessary – all additional parameters are soil type specific and only used in the silting up model
- Silting-up can be used land use specific: When tillage days are given in the land use table, then silting-up will be calculated, otherwise the silting-up-scheme is skipped.

2.13.1.2 surface routing

- designed for small scale applications, otherwise it's to physically based (applications are still

possible, but the interpretation will become more difficult)
- infiltration excess of soil- and silting-up model and direct runoff from snow melt is used as input for the surface routing model
- surface runoff will follow the flow paths from cell to cell until a river cell is reached (note: at least one river cell is required in each subbasin - preferably at the lowest point - otherwise no surface runoff will be generated at all)
- in diverging areas, multiple flow paths (up to three) are possible (if plan curvature < 0, flow paths are diverging, otherwise converging) → use method 1 or 3
- in converging areas, only one flow direction is possible (following the steepest slope)
- flow velocity depends on slope, roughness and water film depth (Manning Strickler approach)
- runoff is assumed to flow as a consistent layer → however, a concentration factor may be used to take into account small scale variations of the surface leading to preferred flow paths in small channels
- slope takes into account the elevation model plus the water storage, thus a dynamic generation of ponds is possible (if elevation + local storage leads to a zero or inverted gradient) → it is still not a hydrodynamic wave approach but rather a cinematic wave approach with step-wise constant parameters. This option is available using methods 3 and 4, only. Methods 1 and 2 use constant slopes without regarding water levels
- roughness can be estimated by different approaches for different types of cultivation
    - fix value for uncultivated areas (e.g. forest)
    - land use dependent parameters for conserving cultivation
    - dynamically estimated parameters for flown through vegetation (originally designed for non-conserving cultivation) taking into account vegetation specific parameters in their annual phenological course and other hydraulic properties
- time step of surface routing is dynamically decreased, if necessary (can be parameterized)
- surface storage of non discharged water is input into the silting-up model or the soil model (depending of configuration) in the next time step
- interflow and base flow are passed through from unsaturated zone model (soil model)
- surface runoff and interflow can optionally be lead into a single reservoir cascade to model the translation and transforming effects of rivers within a subbasin → the same approach as used in the soil model when no surface routing is active. The difference between model run with and without surface runoff is, that for model runs with surface discharge routing only the discharge from cells with a river on it flows into the reservoir cascade. To disable the reservoir cascade, the cell values of the flow time grid may be set to 1 (second) to make sure that all runoff counts for the actual interval only (and the storage recession parameter must be set to a very short time like 0.0001 h).
- total runoff if calculated by adding the subbasin-wide averages of all three runoff components
- total runoff is then input into routing model
- tracer mixing is fully integrated into surface routing

2.13.2 Functional design

Figure 2.13.2.1 outlines the previous scheme for water flows (including tracers). If surface routing and silting-up are switched off, then the new WaSiM version 8.1.01 is functionally identical to the previous versions, so the scheme is still valid in new versions for that case.

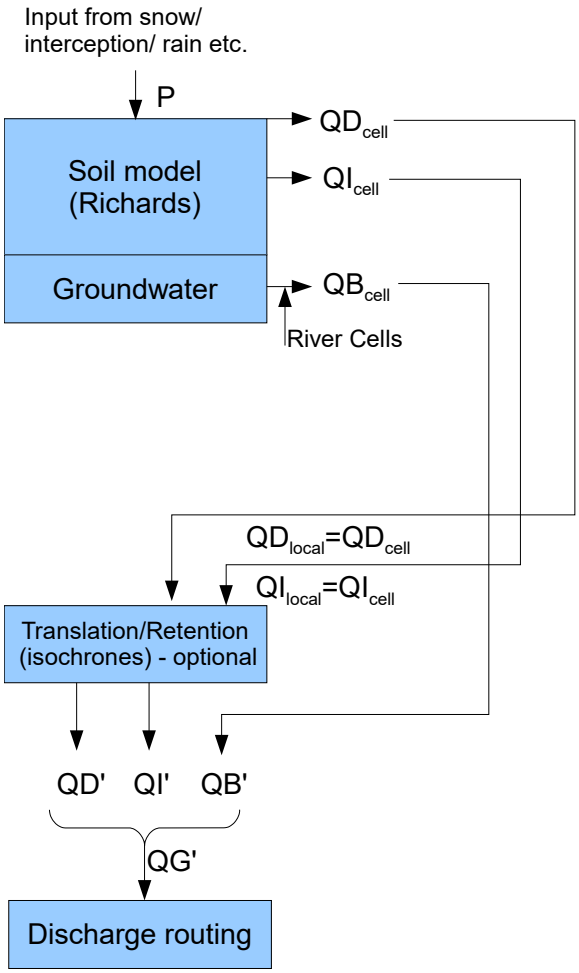


figure 2.13.2.1: Flow chart for traditional soil- and routing modelling (prior to version 8.1.01 and without silting-up and surface routing since version 8.1.01)

As shown, the soil model generates basically three runoff components (drainage runoff is handled as part of interflow and thus not explicitly shown here): direct discharge (QD), interflow (QI), and base flow (QB). If a flow time grid was read in, then QD and QI are transformed separately by a single reservoir cascade (isochrone approach with additional retention in each class). The average values of each runoff component QB', QI', and QD' are summed up to the total runoff QG' which is then the input to the routing model.

Starting with version 8.1.01, two new components may be used to model the generation and transformation of surface runoff. Figure 2.13.2.2 shows the flow chart for the complete scheme. The silting-up module simply separates a fraction of the incoming water (usually precipitation) and stores it as direct discharge QD_{SU} (SU standing for Silting Up). The following Richards scheme is identical to the previous existing one, except for adding the direct discharge generated within the soil model to the already generated direct discharge from the silting-up model. Both components together are written as direct discharge statistics file or grid of the soil model. Note: when using surface routing, the QD statistics file and grids will contain the surface storage for direct runoff, so QD from soil model cannot be used for balancing anymore → use QD_{SR} instead, the statistics output from SurfaceRouting model, or, if no surface routing is active, use QD from soil model as usual.

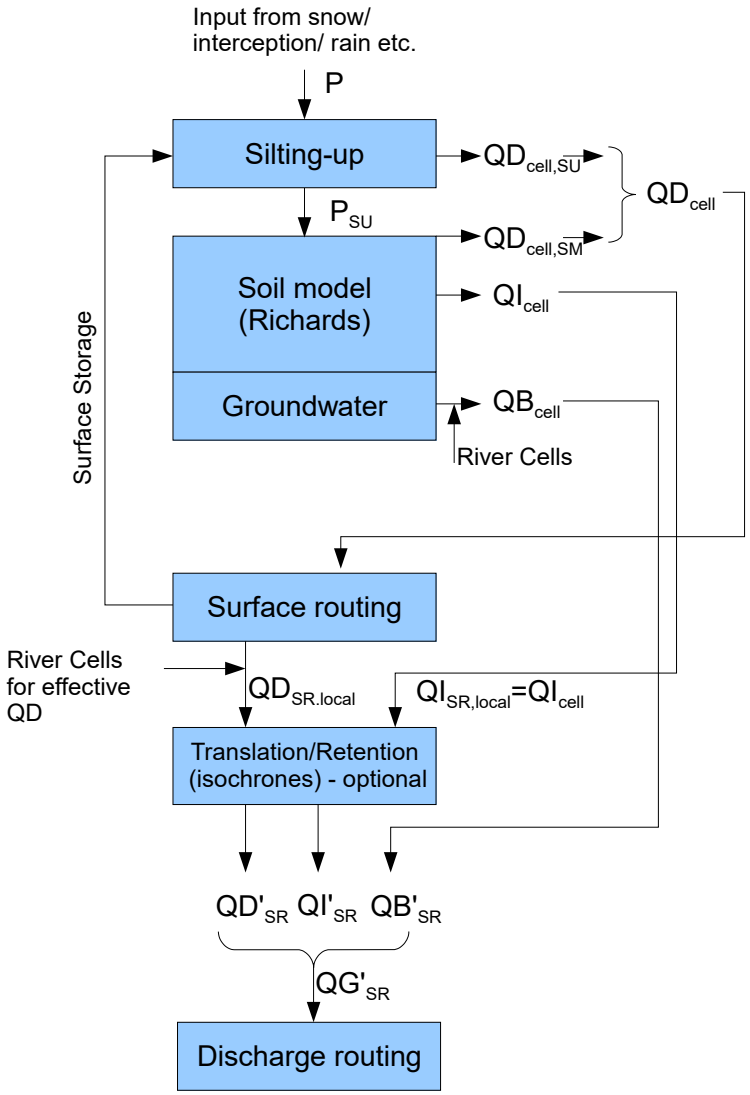


figure 2.13.2.2: Flow chart for soil- and routing modelling including silting-up and surface routing (starting from version 8.1.01)

The other statistics files of the soil model will be written as usual, however: the valid input for balances are the statistical files of the surface routing model. Again: the direct runoff from soil model is NOT valid for balancing when surface routing is used, since it contains the surface storage of the silting up model. The surface routing model fits between the soil model and the (optional) translation/retention of the surface runoff and interflow using two single linear reservoir cascades. Surface runoff on a cell will only count as surface runoff for the statistics, if the cell is a valid river cell. All other cells will store the routed surface runoff for further routing until a river cell is reached. This implies the presence of a river network grid with at least on valid river cell per subbasin, otherwise no surface runoff will leave the subbasin at all. Interflow and baseflow are transferred untouched to the discharge routing module, so these two statistic files will be identical. The resulting total runoff QG'_{SR} is calculated as the sum of the routed surface runoff QD'_{SR} (optionally transformed by the single linear reservoir [SLR] cascade), QI'_{SR} (as well optionally transformed by a SLR-cascade), and QB'_{SR}. As described in the overview, this reservoir cascade can be virtually disabled by setting all cells in the flow time grid to 1 (so only one flow time zone will be created) and setting the single linear reservoir recession constants for direct and/or interflow to a very small value. It is not possible to switch the reservoir cascade off for one flow component only. However, the direct runoff will enter the reservoir cascade only when it is routed on the surface and reaches a river cell. Thus, the effect of the reservoir cascade is rather small if there are only a few

river cells at the outlet of the sub-catchments.

Between the two extremes of using no surface runoff and no silting up on one side and using both modules on the other side, there are two more possibilities: both modules could be used without the other module, so a scheme as shown in Figures 2.13.2.3 and 2.13.2.4 is perfectly possible.

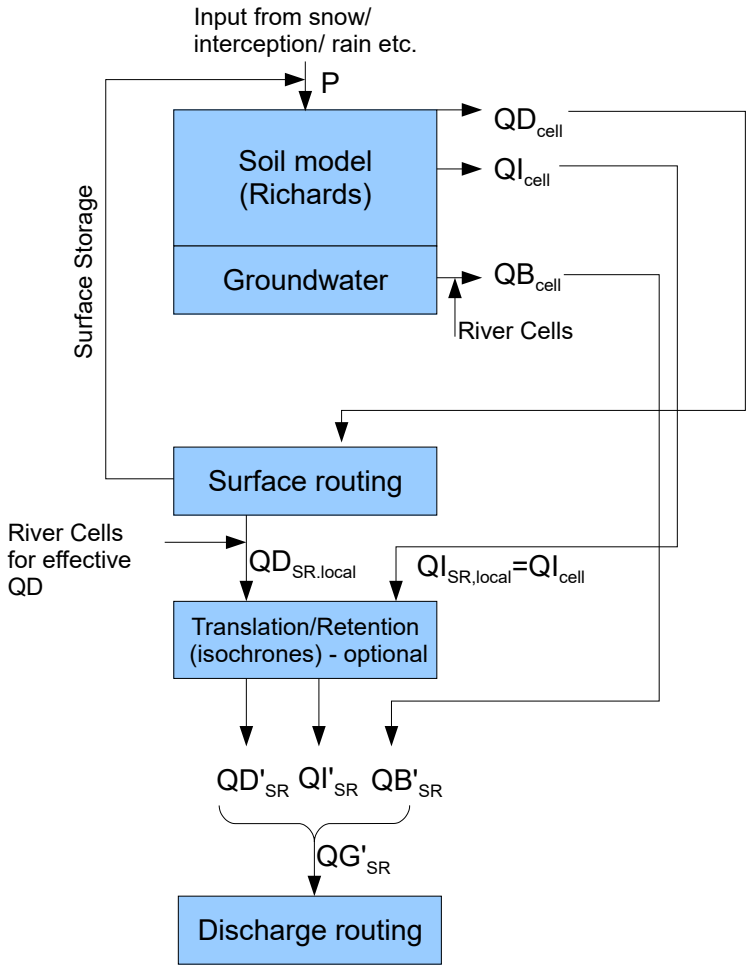


figure 2.13.2.3: Flow chart for soil- and routing modelling including surface routing but without silting-up

If no silting-up model is used, then the input into the soil model is the traditional input from the interception storage or snow model or precipitation. Since no other surface runoff component is calculated, the entire surface runoff is generated in the soil model (exception: surface runoff from snow may be there, but this is calculated in the soil model as well). All other flows are calculated as described before. The surface routing storage which is used in both models, surface routing and silting-up, is used in the surface routing model only.

Figure 2.13.2.4 delineates the application of the silting-up model *without* using the surface routing model. Again, the surface routing storage is used by one model component only (surface routing and silting-up module will check during initialization if the surface routing storage already exists and create it, if required).

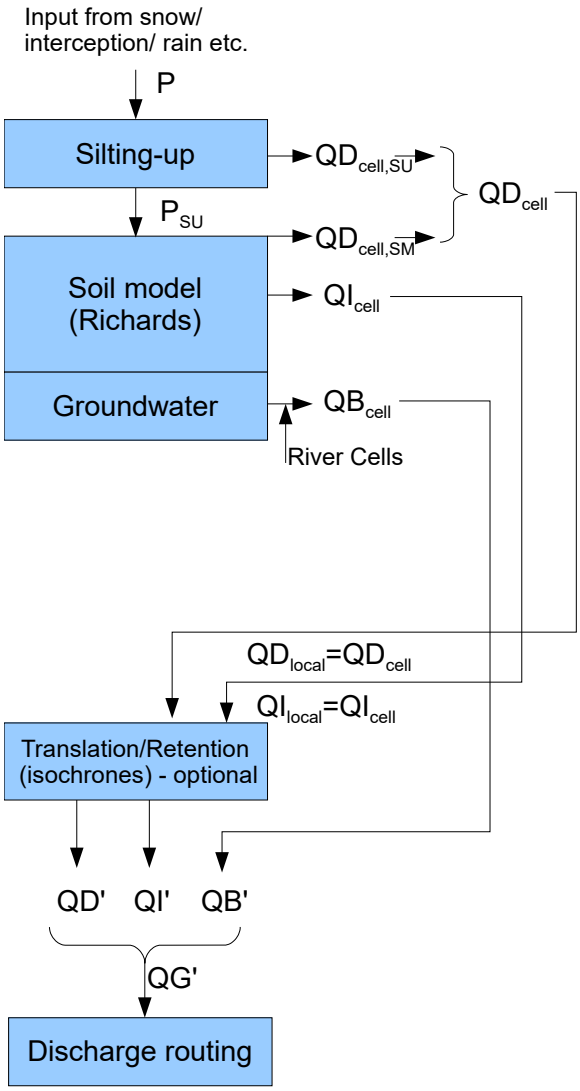


figure 2.13.2.4: Flow chart for soil- and routing modelling including silting-up but without surface routing

All fluxes from the soil model are identical to the traditional model scheme, except for the two surface runoff components which are internally added to a total surface runoff.

2.13.3 Theory of the Silting-up module

The core equation to estimate the potential infiltration rate is given in equation (2.13.1) :

$$i_{pot}(t) = (i_0 - i_\infty) \cdot e^{-C_v \cdot E_{kin}(t)} + i_\infty \qquad \textbf{(2.13.1)}$$

with $i\text{pot}(t)$ potential infiltration rate at time t [mm/h]
i_0 start infiltration rate (fix at 65 mm/h)
i_∞ end infiltration rate (see below) [mm/h]
C_v silting-up disposition depending on grain size distribution (see below) [m^2/J]
$E_{kin}(t)$ measure for cumulative kinetic energy at time t (see below) [J/m^2]

The total potential infiltration for a model time step is calculated after:

$$I_{pot}(t) = i_{pot}(t) \cdot \Delta t$$

(2.13.2)

with I_{pot}(t) potential infiltration [mm]
Δt model time step [h]

Applied to the actual input into the cell consisting of precipitation and already present water from inflow or local storage, the actual infiltration can be calculated as:

$$I_{act}(t)=p(t)+q_{in}(t) \quad \text{if} \quad I_{pot}(t)>\left(p(t)+q_{in}(t)\right)$$
$$I_{act}(t)=I_{pot}(t) \quad \text{if} \quad I_{pot}(t)\leq\left(p(t)+q_{in}(t)\right) \qquad \textbf{(2.13.3)}$$

with $I_{act}(t)$ actual infiltration for time step t [mm]
$p(t)$ precipitation (input) into the cell for time step t [mm]
$q_{in}(t)$ inflow from other cells and local storage from last time step [mm]

The surface runoff for the actual time step $QD_{SU}(t)$ results to:

$$QD_{SU}(t)=p(t)+q_{in}(t)-I_{act}(t) \qquad \textbf{(2.13.4)}$$

To this infiltration excess additional surface runoff may be added by the Richards approach in the unsaturated zone model. Thus, the actual infiltration is in fact the minimum of the infiltration rates calculated by Richards approach and by equations (2.13.1) to (2.13.3) .

There are some entities from equation (2.13.1) which must be estimated or parameterized separately.

The end infiltration rate i_{∞} is calculated by:

$$i_{\infty}=12.2\cdot d_g^{0.52}\cdot f_d^{-0.64} \qquad \textbf{(2.13.5)}$$

with d_g geometric mean of grain size distribution [µm] (see below)
f_d fraction of sand [weigth-%]

$$d_g=10^{\frac{c_y\cdot f_y+c_t\cdot f_t+c_d\cdot f_d+c_{s1}\cdot f_{s1}+c_{s2}\cdot f_{s2}+c_{s3}\cdot f_{s3}+c_{s4}\cdot f_{s4}}{200}} \qquad \textbf{(2.13.6)}$$

with f_y fraction of clay [weight-%]
f_t fraction of silt [weight-%]
f_d fraction of sand [weight-%]
f_{s1} fraction of stones 2 - 6.3 mm [weight-%]
f_{s2} fraction of stones 6.3 - 20 mm [weight-%]
f_{s3} fraction of stones 20 - 63 mm [weight-%]
f_{s4} fraction of stones 63 - 200 mm [weight-%]
→ all values of f_y ... f_{s4} are given as relative values [-] in the control file
c_y...c_{s4} weighting factors for the respective f_y..f_{s4} defined as follows:

$$c_y=\log_{10}(0.04)+\log_{10}(2)=-1.0969 \qquad \textbf{(2.13.7)}$$

$$c_t=\left\{\begin{matrix}0.3326\left(\log_{10}2+\log_{10}6.3\right)+0.3348\left(\log_{10}6.3+\log_{10}20\right)\\+0.1704\left(\log_{10}20+\log_{10}36\right)+0.1622\left(\log_{10}36+\log_{10}63\right)\end{matrix}\right\}=2.1004 \qquad \textbf{(2.13.8)}$$

$$c_d = \left\{ \begin{array}{l} 0.1336\left(\log_{10}63+\log_{10}100\right)+0.2005\left(\log_{10}100+\log_{10}200\right) \\ +0.3318\left(\log_{10}200+\log_{10}630\right)+0.3341\left(\log_{10}630+\log_{10}2000\right) \end{array} \right\} = 5.1004 \quad \textbf{(2.13.9)}$$

$$c_{s1} = \left(\log_{10}2000+\log_{10}6300\right) = 7.1004 \quad \textbf{(2.13.10)}$$

$$c_{s2} = \left(\log_{10}6300+\log_{10}20000\right) = 8.1004 \quad \textbf{(2.13.11)}$$

$$c_{s3} = \left(\log_{10}20000+\log_{10}63000\right) = 9.1004 \quad \textbf{(2.13.12)}$$

$$c_{s4} = \left(\log_{10}63000+\log_{10}200000\right) = 10.1004 \quad \textbf{(2.13.13)}$$

The silting-up disposition C_v is calculated by:

$$C_v = 0.013 \cdot f_d^{-1.03} \cdot d_g^{0.7} \cdot t_{cult}^{-0.19} \quad \textbf{(2.13.14)}$$

with f_d fraction of sand [%] as above
f_y fraction of clay [%] as above
t_{cult} time since last soil cultivation (min(t_{cult})=0.3) [d]

If silting up is calculated on a day with soil cultivation, the value of t_{cult} is set to 0.3 (not 0). If no tillage days are specified in the control file, then no silting-up will be calculated for this land use at all.

The cumulative kinetic energy $E_{kin}(t)$ is calculated after the following scheme:

$$\begin{array}{lll} e_n(t)=0 & \text{if} & n(t)<0.05 \\ e_n(t)=28.33 & \text{if} & n(t)>76.2 \\ e_n(t)=11.89+8.73\cdot\log_{10}\left(n(t)\right) & \text{if} & 0.05\le n(t)\le 76.2 \end{array} \quad \textbf{(2.13.15)}$$

with $e_n(t)$ energy of precipitation for time step t [J/(m^2·mm)]
$n(t)$ precipitation intensity [mm/h]

$$\begin{array}{lll} E_{kin}(t)=e_n(t)\cdot n(t)\cdot\left(1-f_{res}(t)\right)\cdot\Delta t & \text{if} & t_{RP}>t_{RP,min} \\ E_{kin}(t)=E_{kin}(t-\Delta t)+e_n(t)\cdot n(t)\cdot\left(1-f_{res}(t)\right)\cdot\Delta t & \text{if} & t_{RP}\le t_{RP,min} \end{array} \quad \textbf{(2.13.16)}$$

with $E_{kin}(t)$ cumulative kinetic energy of precipitation at time t [J/m^2]
f_{res} fraction of soil covered by residues, plants and stones (total) [1/1]
Δt duration of a time step [h]
t_{RP} rain pause (time since last precipitation with no inflows and local storage = 0) [d]
$t_{RP,min}$ minimum rain pause (value given in control file for each subbasin) [d]

2.13.4 Control file extensions for Silting-up module

Starting with WaSiM 8.01.01, a new submodel parameter set named `[SiltingUpModel]` is added to the control file, which must be present in any control file (if silting-up is not used: at least the name of the submodel and the switch-parameter set to 0 must be present).

The example below shows a possible parameterization. The grid containing the surface water storage is shared between the two submodels “silting-up” and “surface routing”. It is created internally. To be able to write this grid to disk (which may be a good idea, especially when coupling different model runs, since the surface storage must be read in at the start of the coupled model run), the surface storage grid should be defined as variable grid with a default value of 0 (and a replace-

nodata-value of 1).

The meaning of the parameters is described in the following example. Minimum rain pause must be defined for each subbasin. However, the abbreviation as commonly used in [unsatzonmodel] can be used here, too: omitted subbasin codes and parameter values are filled in automatically by the control file reader. So, a unique value for all subbasins in the example could be defined be simply set only the value for subbasin 1 (subbasin 1 must exist, however).

```
$set $SurfStorSiltingUp    =  sfstsu//$grid//.//$suffix
$set $SDISPgrid            =  sdis//$grid//.//$suffix
$set $RPAUSgrid            =  paus//$grid//.//$suffix
$set $EKIN_grid            =  ekin//$grid//.//$suffix
$set $TSBB_grid            =  tsbb//$grid//.//$suffix
$set $QDSU_grid            =  qdsu//$grid//.//$suffix

→ optional entry in variable grids: if the grid is not listed here, it is created internally and cannot be written
[variable_grids]
...
$outpath//$SurfStorSiltingUp  SurfStorSiltingUp  1  0  # storage for surface runoff
$Writegrid                                             # Writegrid for this grid
$readgrids                                             # 0 or 1
...

[SiltingUpModel]
1                           # 0=ignore this module, 1 = run the module
$time                       # duration of a time step in minutes
$outpath//sdis//$grid//.//$year $hour_mean # statistics for silting up disposition $outpath//qdsu//$grid//.//$year
$hour_mean # direct discharge from silting up module
$outpath//$SDISPgrid        # grid with actual silting up disposition
$Writegrid                  # writegrid for this grid
$outpath//$RPAUSgrid        # grid with actual rain pause length
$Writegrid                  # writegrid for this grid
$outpath//$EKIN_grid        # grid with actual cumulated kinetic energy of the event
$Writegrid                  # writegrid for this grid
$outpath//$TSBB_grid        # grid with actual time since last soil tillage
$Writegrid                  # writegrid for this grid
$outpath//$QDSU_grid        # grid with direct runoff from silting up model
$Writegrid                  # writegrid for this grid
1    2    3    4    5    6    7    8    9    10   11   12   13  # range for subbasin codes
1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0 # minimum Rainpause (in days)
$readgrids                  # readgrid code 0 do not read, 1 = read grids
```

Also, the land use table should contain the days of soil cultivation:

```
6 agriculture     {method    = VariableDayCount;
                 SoilTillage      = 90 240;     # optional set of 1..n Julian days, depicting days
                                                  with soil tillage. Important for silting up model
                  SR_ResidCovrg = ...see surface routing below, this parameter will be used in
                                   silting-up-model, too
....
```

If the model starts prior to the first tillage day, then the last tillage day is assumed to be the same JD in the last year, so in the above example, the last soil tillage has been on day 240 in the old year (which was not modelled), so the time since last tillage is 365-240 = 125 plus the actual Julian day (126 for day one).

The last required control file extension is a set of grain size distribution values which should be added to each soil type in the soil table.

The specific parameter is called GrainSizeDist and consists of 7 values:

- volumetric fraction of sand (default = 0.3333)
- volumetric fraction of silt (default = 0.3333)
- volumetric fraction of clay (default = 0.3334)
- volumetric fraction of stones 2-6.3 mm (default = 0)
- volumetric fraction of stones 6.3-20 mm (default = 0)
- volumetric fraction of stones 20-63 mm (default = 0)

- volumetric fraction of stones 63-200 mm (default = 0)

```
1 sand_(S) {method = MultipleHorizons;
       FCap = 6.21; mSB = 38.5; ksat_topmodel = 8.25E-5; suction = 385; # optional for Topmodel
        GrainSizeDist  = 0.75 0.1 0.05 0.05 0.03 0.01 0.01; # optional: when using silting up
model, #the grain size fractions for sand, silt, clay, and Stones1..4  must be given here.
#Stones1 = 2-6.3mm, Stones2=6.3-20mm, Stones3=20-63mm, Stones4=63-200mm.
       PMacroThresh   = 1000;  # precipitation capacity thresholding macropore runoff in mm/hour
...
```

2.13.5 Theory of the Surface routing module

Note: there is a new extension starting with version 10.00.00 that uses a system of linear equations similar to the groundwater model, resulting in much smoother lake surfaces and using much less iteration steps. See chapter 2.13.7 .

Within the surface routing module we try to apply a kinematic wave approach for unsteady conditions including backwater effects. This would be basically impossible if we would not use a quasi-stationary approach. The basic concept is this:

- We divide the time step in as much sub steps as required to approximate unsteady conditions as good as possible
- One important parameter for an automated subdivision is the minimum flow time, the water needs from one cell to the next cell. The internal time step should not exceed this time since the inflow/outflow balancing only accounts for direct neighbors.
- When strong backwater conditions exists (very flat regions with shallow gradients but a lot of water flowing into such regions), the sub time step may be manually limited to force the model to work with smaller steps
- For each sub time step, the parameters like gradient and roughness and the inflow data are newly calculated. Then, the fluxes are routed.
- Using this scheme, it is well possible that the gradient changes from one sub time step to the next, so the next sub time step will result in a different flow velocity and/or flow direction (or even in a reversed flow direction). To minimize numerical oscillations, the sub time step has to be sufficiently small compared to the normal time step. So it may be a good idea to set the maximum allowed sub time step to $1/4^{th}$ to $1/10^{th}$ of the normal time step or even shorter.

The following explanations describe the scheme, WaSiM uses to implement this concept including the used equations (Source code encompasses appr. 2500 lines of C++-Code).

1. First of all, the input for each raster cell is taken from the preceding modules. This is usually the surface runoff calculated in the unsaturated zone model and/or in the silting-up model. Since the input is given as a total amount for the entire time step without differentiation between newly generated surface runoff (from infiltration excess or exfiltrated groundwater) and water which was already stored on the land surface from preceding time steps, it is not evenly distributed to all sub time steps but used as bulk input. This means, that there will be a tendency for higher water levels in the beginning of an event. But the flow velocities of runoff generated by from rain-runoff only are rather small compared to the velocities, stored water will allow (because of the much deeper water bodies). The disadvantage of the bulk input will be small compared to the numerical simplification.

   The input is defined as $Q_{in,total}$, given in mm per time step.

2. Now, the flow velocities must be estimated. This is a rather complex task, since it incorporates roughness estimation, slope and aspect calculation (all dynamically!) and

multiple flow paths definition (among others).

The slope or gradient will be determined for each cell as maximum positive difference between the elevations of the central cell and the eight neighboring cells (with a reduction factor of $\sqrt{2}$ = 1.414 for diagonal directions). Directions in the following equations are coded as:

North	=	1
North-East	=	2
East	=	3
South-East	=	4
South	=	5
South-West	=	6
West	=	7
North-West	=	8

The slope is calculated by:

$$S = \underset{i=1}{\overset{8}{Max}} \left(\frac{(h_0 + h_{w,0}) - (h_i + h_{w,i})}{c_s \cdot f_i} \right) \qquad \textbf{(2.13.17)}$$

with S slope [-]
h_0 absoulte elevation of the actual cell from the digital elevation model [m]
$h_{w,0}$ water level ($Q_{in,total}$ /1000) for the actual central cell [m]
h_i absolute elevation from digital elevation model for the neighboring cell I with i as a code for the direction as given above
$h_{w,i}$ water level ($Q_{in,total}$ /1000) for cell i [m]
c_s cell size [m]
f_i direction dependent weighing factor:
orthographic directions: 1.0
diagonal directions: 1.4142 = $\sqrt{2}$

The values for $h_{w,0}$ and $h_{w,i}$ are only used when the method is set to 3 (multiple plow paths regarding the water level) or 4 (single flow path regarding water level). Methods 1 (multiple flow paths without regarding water level) and 1 (single flow paths without regarding water level) will set both values to 0.

The aspect or direction of exposition Ω [° from north] is defined by the direction of the steepest slope.

Special cases:

a) multiple slopes with the same value:

Although this case will be rare since even the slightest difference in water levels will give a unique main flow direction for methods 3 and 4 , it may occur when using methods 1 or 2. The maximum slope S can still be calculated, but the aspect must be identified differently. Here, the approach used for the topographic analysis is used:

$$\Omega = 180 - \arctan(H/G) + 90(G/|G|) \qquad \textbf{(2.13.18)}$$

with Ω aspect angle, rounded to the nearest multiple of 45° (matching one of the directions 1...8 coded above)
G South-North-gradient (h_5 - h_1)/2 c_s [m]
H West-East-gradient (h_7 - h_3)/2 c_s [m]

b) no slope at all:

S will be set to 0, *Ω* will be set to 0 (north, direction code 1)

c) missing cells at the border: the actual grid cell is at the border of the model domain (nodata-values or no values at all are partly surrounding the actual cell): Equation (2.13.17) is applied for valid surrounding cells only . If special cases a) or b) happen, they are solved as described above. Missing elevations for equation (2.13.18) are taken from the central cell, if required, with an additional difference regarding the gradient from the central cell to the cell in the opposite direction (e.g., when no cell to the north is present but to the south, the elevation of the northern cell is estimated by taking the elevation of the central cell and adding the elevation difference between the central and the southern cell; same for western or eastern cell).

When knowing the main flow direction (aspect angle in 45° steps) and its gradient, the multiple flow paths can be identified. WaSiM detects multiple flow path based on two assumptions:

a) the plan curvature of the actual cell is negative → the flow is diverging

b) multiple flow paths are possible to the left and to the right of the main flow direction only (in 45°-steps). Exception: if gradients to all directions are identical, then flow into all directions is possible.

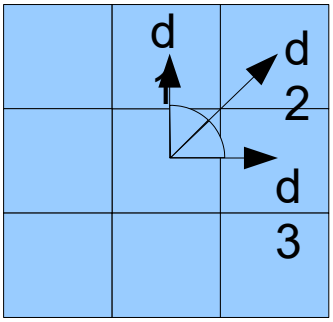
The plan curvature ω is calculated by:

$$\omega = 2\frac{DH^2 + EG^2 - FGH}{G^2 + H^2} \qquad \textbf{(2.13.19)}$$

with D $((h_1 + h_5)/2 - h_0)/c_s^2$ see equation (2.13.17)
E $((h_3 + h_7)/2 - h_0)/c_s^2$ see equation (2.13.17)
F $(h_2 + h_6 - h_4 - h_8)/4\,c_s^2$ see equation (2.13.17)
G $(h_5 - h_1)/2\,c_s$
see equation (2.13.17)
H $(h_7 - h_3)/2\,c_s$ see equation (2.13.17)

If the plan curvature is positive, the main flow direction will be the only flow path for that cell. For negative plan curvatures, the flow paths 45° to the left and 45° to the right are equally weighted. They are checked for a positive gradient. If the gradient of one or both minor flow directions is negative or 0, then the respective flow direction weighting is set to 0. The following drawing shows the main principles: d2 is the main flow direction with weight $w_2 = 1.0$. d1 and d3 are minor flow directions with weights $w_1 = w_3 = 0.7071$, according the *cos* (45°). All weights are normalized to $w_1 + w_2 + w_3 = 1.0$, so the resulting weights are

$$w_1 = 0.2929$$
$$w_2 = 0.4142 \qquad \textbf{(2.13.20a)}$$
$$w_3 = 0.2929$$

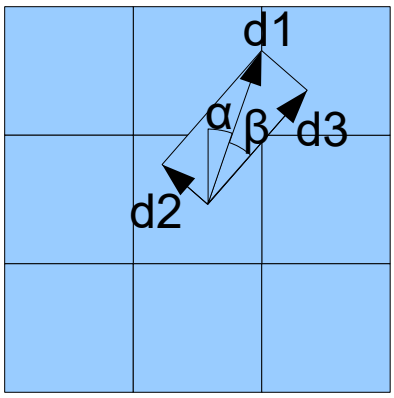


For aspects not fitting into the 45°-raster (i.e. calculated by eq. (18)), the weights w1 to w3 are calculated by:

$$
\begin{aligned}
w_1 &= \cos(\alpha)\cdot\cos(\beta) \quad \text{if} \quad \alpha \geq 0 \\
w_1 &= \cos(\alpha)\cdot\sin(\beta) \quad \text{if} \quad \alpha < 0 \\
w_2 &= \cos(\alpha) \\
w_3 &= \cos(\alpha)\cdot\sin(\beta) \quad \text{if} \quad \alpha \geq 0 \\
w_3 &= \cos(\alpha)\cdot\cos(\beta) \quad \text{if} \quad \alpha < 0
\end{aligned}
\qquad \textbf{(2.13.20b)}
$$

with α angle between the main flow direction (d1 in the drawing below) and the nearest flow direction class (1 to 8, as given above), range between -22.5°...22.5°

β complementary angle of α to ± 45° (because the next minor flow direction d2 or d3 – depending on the sign of α - will always have an angle of 45° against the next main flow direction class, see drawing below)



As for 45°-steps-aligned aspects, weights are normalized to 1.

After identification of the flow directions and their weightings, the flow velocities for each direction are calculated. The basic equation is the Manning-Strickler equation:

$$v_i = M \cdot R_H^{2/3} \cdot S_i^{1/2} \qquad \textbf{(2.13.21)}$$

with v_i flow velocity in direction i (as coded above)
M roughness coefficient after Manning-Strickler (see below)
R_H hydraulic radius = water depth
S_i gradient in direction I

Solving equation (2.13.21) is an iterative approach, since R_H, v and partly M depend on each other. So the starting value of v_i will be $v_{i,0}$ = 0.1m/s. The starting value of R_H is set to

$$R_H = \frac{q_i}{v_{i,0} \cdot c_s} \quad \text{with} \quad q_i = \frac{Q_{in,total} \cdot c_s^2 \cdot w_i}{1000 \cdot \Delta t} \qquad \textbf{(2.13.22)}$$

with q_i flow in direction i [m³/s]
$v_{i,0}$ initial flow velocity in direction i [m/s]
w_i weighting of flow in direction i [-] as defined above
Δt time step [s]
$Q_{in,total}$ total input in mm/ Δt

Also the roughness coefficient M may depend on water level, see the explanations below.

The first value of R_H is then used to calculate $v_{i,1}$ which in turn is used for a second estimation of $R_{H,1}$ which is then used for $v_{i,2}$-calculation and so on until one of the iterations break conditions is reached:

$$\left(\left|v_{i,n} - v_{i,n-1}\right| \le \Delta v_{max}\right) \vee \left(n > n_{max}\right) \qquad \textbf{(2.13.23)}$$

where n iteration counter
n_{max} maximum number of iteration steps (parameter from control file)
Δv_{max} maximum difference between two subsequent iteration steps
(n_{max} and Δv_{max} are both parameters from the control file)

Having calculated the flow velocity and the hydraulic radius, two more checks are done. First, the possibility of **preferential flow paths** formation must be checked. This is done by calculating the shear stress and compare it against a given critical shear stress. If the actual shear stress is greater then the critical shear stress, the flow is artificially concentrated by scaling R_H by a factor given in the control file. This changed R_H requires a recalculation of the iterative equations (2.13.21) and (2.13.22) . The shear stress τ [Pa] is calculated by:

$$\tau = \rho \cdot g \cdot R_H \cdot S_i \cdot \left(\frac{M}{M_g}\right)^{3/2} \qquad \textbf{(2.13.24)}$$

with ρ density of water at 20°C, here taken as 1000 [kg/m³]
g gravitational acceleration = 9.815 m/s²
M roughness coefficient as calculated here for certain land uses (see below) in m$^{1/3}$/s
M_g roughness coefficient for bare ground as given in the control file [m$^{1/3}$/s]

The critical value τ_{crit} can be given as cell-by-cell values as a standard grid or, if no such grid is available, is set to 0.9 per default. If $\tau > \tau_{crit}$, R_H is scaled by the concentration factor from the control file and eq.s ((2.13.21) and (2.13.22) are applied for the changed R_H again.

The second check deals with **too slow velocities** and thus with a too large hydraulic radius. If the gradient is very shallow or the flow velocity is very small because of low M-values, the iteration may result in R_H-values which are larger than the total input $Q_{in,total}$ was. Since this is physically impossible, the hydraulic radius must be reduced to the physical maximum. This will lead to a reduced

flow velocity, too, so equations (2.13.21) and (2.13.22) will be applied once more. In this third iteration, the R_H-value is kept constant, so the iteration is usually only a one-step iteration, except when M changes (see below).

The flow velocity is now used to calculate the **effective fluxes** in m^3/s:

$$q_{i,eff} = \frac{v_i \cdot c_s \cdot R_H}{f_{conc}} \qquad \textbf{(2.13.25)}$$

with $q_{i,eff}$ effective flux in direction i [m^3/s]
f_{conc} concentration factor as parameter from the control file (or 1, if no channel generation takes place)

Calculation of roughness coefficient *M* :

There are three different algorithms for estimating *M*:

- **Roughness coefficient *M* for other land use than cultivated (e.g. Forest)**

  M is simply taken from the control file als M_g (g=ground).

- **Roughness coefficient *M* for conserving land use:**

  It is assumed, that the crop parameters are negligible for the total roughness. Conserving cultivation doesn't use ploughs and the residues remain on the field (or mulch will even be thrown out). *M* depends on the flow velocity v_i and on the hydraulic radius R_H. Starting point is the Reynolds number, a measure for the intrinsic friction due to kinematic viscosity of water:

$$Re = \frac{4 \cdot v_i \cdot R_H}{\nu} \qquad \textbf{(2.13.26)}$$

with Re Reynolds number
ν kinematic viscosity at 20°C = $1.0 \cdot 10^{-6}$ m^2/s
R_H hydraulic radius [m], $4 \cdot R_H$ = characteristic length for Re

The roughness parameter M is then calculated by:

$$n_r = \frac{a \cdot f_{veg}^b}{Re^c} \quad \rightarrow \quad M = \frac{1}{n_r} \qquad \textbf{(2.13.27)}$$

with n_r roughness coefficient [$s/m^{1/3}$]
a,b,c crop specific coefficients [-]
f_{veg} fraction of covered soil [%]

As can be seen by equations (2.13.26) and (2.13.27) , the ground roughness M_g has no effect on M here. Only the plant covered fraction (ideally: the fraction covered by residues) of the soil (taken from land use table) and the crop specific coefficients are parameters for M. The three empiric parameters a, b and c are given in Table 2.13.1 (G ILLEY):

Table 2.13.1 : coefficients for calculating roughness coefficient M for conserving cultivation

	***Re* < 20000**			***Re* ≥ 20000**		
Crop type	**a**	**b**	**c**	**a**	**b**	**c**
corn/maize	0.00496	0.89200	0.03110	0.00519	1.20000	0.17700
cotton	0.00896	0.67800	0.00930	0.00473	0.70000	-0.03260
peanuts	0.02730	0.70300	0.19100	0.00773	0.41100	-0.05090
pine(-needles)	0.00539	1.04000	0.19200	0.000332	1.23000	0.00311
sorghum	0.07850	0.48800	0.01980	0.03	0.71400	0.18900
soybeans	0.04510	0.91300	0.35800	0.00159	0.96100	0.05100
sunflower	0.04130	0.43900	0.08930	0.00411	0.85800	0.03300
wheat	0.00207	1.46000	0.30200	0.000192	1.03000	-0.09780
others	0.01890	0.71200	0.14200	0.01890	0.71200	0.14200

The coefficients given in table 2.13.1 must be set in the control file per land use type. Default values (if omitted) are the parameters for crop type „others“.

- **Roughness Coefficient *M* for flow through vegetation (orig.: non-conserving land use)**

  Non-conserving cultivation is conventional cultivation using ploughs. However: This approach should mainly be used for flowed through vegetation (bushes, woods etc.). M is assumed to be a result of the grounds roughness and the roughness of vegetation, the water flows through. The approach is based on the algorithms suggested by PASCHE , an improvement of the approach by LINDNER . Vegetation roughness is effectively used for flow depths above a specified value (in the control file). This is because of instabilities of the algorithms for very shallow water depths.

  Ground roughness is calculated by

$$\lambda_g = \frac{8 \cdot g}{R_H^{1/3} \cdot M_g^2} \qquad \textbf{(2.13.28)}$$

  with λ_g roughness of the bare ground [-]
  M_g Manning-Strickler coefficient for bare ground, given in the control file [$m^{1/3}/s$]
  R_H hydraulic radius [m]
  g gravitational constant = 9.81 m/s^2

  Vegetation roughness is calculated by the following scheme.

  At first, a few parameters will be estimated:

$$Re_{veg} = \frac{v \cdot d_p}{\nu} = \frac{v \cdot d_p}{10^{-6}} \qquad \textbf{(2.13.29)}$$

  with Re_{veg} Reynolds number for flowed through vegetation [-]
  v flow velocity (iterative) [m/s]
  d_p stem diameter of the flowed through vegetation
  ν kinematic viscosity of water at 20°C (=10^{-6}) [m^2/s]

The form drag coefficient of a single circular cylinder is derived from Re $_{veg}$:

$$\begin{aligned} c_{w\infty} &= 3.07 \cdot Re_{veg}^{-0.168} \quad \text{if} \quad Re_{veg} < 800 \\ c_{w\infty} &= 1.0 \quad \text{if} \quad 800 \le Re_{veg} < 8000 \\ c_{w\infty} &= 1.2 \quad \text{if} \quad 8000 \le Re_{veg} < 10000 \end{aligned} \qquad \textbf{(2.13.20)}$$

where $c_{w,\infty}$ form drag coefficient of a single circular cylinder [m]

For Re $_{veg}$ > 10000, the value of cw from the control file is used instead (form drag coefficient for a circular cylinder in a group), default value is around 1.5

Two more intermediate entities are now estimated: the wake length a $_{WL}$ and the wake width a $_{WW}$ characterizing the dimensions of the wake behind a circular cylinder:

$$a_{WL} = 128.9 \cdot c_{W\infty} \cdot d_p \cdot \left(1 + \frac{g \cdot a_{WL} \cdot I_E}{v^2/2}\right)^{-2.14} \qquad \textbf{(2.13.31)}$$

with a $_{WL}$ wake length [m]
I $_E$ gradient (usually energy curve gradient, but here set to ground slope)
v flow velocity

Since a $_{WL}$ is dependent on itself, equation (2.13.31) can only be solved iteratively. Starting parameters are a $_{WL,1}$ = 50·d $_p$, and Δa $_{WL}$ = 1.0. The control file contains two parameters for controlling the iteration: maximum wake length iteration difference Δa $_{WL,max}$ (e.g. 0.001m) and the maximum number of iterations, e.g. 40.

As some analyses have shown, the iteration may not always converge. However, WaSiM needs to have stable results. To ensure this, the effective gradient is limited:

$$\frac{c_{w\infty} \cdot I_E}{v^2} > 0.04\,? \quad \rightarrow \quad I_E = \frac{v^2}{25 \cdot c_{w\infty}} \qquad \textbf{(2.13.32)}$$

Now, a $_{WW}$ can be identified:

$$a_{WW} = 0.24 \cdot a_{WL}^{0.59} \cdot \left(c_{W\infty} \cdot d_p\right)^{0.41} \qquad \textbf{(2.13.33)}$$

with a $_{WW}$ wake width [m]

To calculate the form drag coefficient of the vegetation c $_{WR}$ (not to be taken for $c_{w,\infty}$, the form drag coefficient for a single uninfluenced circular cylinder!), the relative flow velocity as relation between the velocities before and after a cylinder is estimated, using a $_{WL}$ and a $_{WW}$:

$$\left(\frac{v_b}{v_a}\right)^2 = 1.15 \cdot \left(\frac{a_{WL}}{a_x}\right)^{-0.48} + 0.5 \cdot \left(\frac{a_{WW}}{a_y}\right)^{1.1} \qquad \textbf{(2.13.34)}$$

with v $_b$ flow velocity before the obstacle [m/s]
v $_a$ flow velocity after the obstacle [m/s]
a $_x$ interspace between crops in flow direction, from control file [m]
a $_y$ interspace between crops perpendicular to flow direction, from control file [m]

The last required parameter for calculating c $_{WR}$ is Δc $_W$, a correction term accounting for

scaling effects which depends on the Froude number Fr:

$$Fr = \frac{v}{\sqrt{(g \cdot R_H)}} \qquad \textbf{(2.13.35)}$$

with Fr Froude number: $Fr < 1$: laminar flow, $Fr > 1$: turbulent flow [-]

$$\Delta c_W = \frac{2}{Fr^2}(1-y) \quad \text{with} \quad y = 2 \cdot Fr \cdot \frac{\left(y - \frac{a_y}{a_y - d_p}\right)}{y^2 - 1} \qquad \textbf{(2.13.36)}$$

with y internal intermediate result

The identification of y is again an iterative process (using 0.0001 as maximum iteration error and taking the maximum number of iterations from the same parameter as the a_{WL}-iteration). To ensure the iterations convergence, some checks and adjustments are required:

$$a_y < 5 \cdot d_p ? \quad \rightarrow \quad a_y := 5 \cdot d_p \qquad \textbf{(2.13.37)}$$

$$Fr \cdot \left(\frac{a_y}{a_y - d_p}\right)^3 \geq \frac{2}{3} ? \quad \rightarrow \quad Fr = \frac{2}{3 \cdot \left(\frac{a_y}{a_y - d_p}\right)^3} \qquad \textbf{(2.13.38)}$$

$$Fr < 0.05 ? \quad \rightarrow \quad Fr = 0.05 \qquad \textbf{(2.13.39)}$$

The value of Δc_W is then calculated after equation (2.13.36) (first part).

Now c_{WR} can be calculated by:

$$c_{WR} = 1.31 \cdot c_{W\infty} \cdot \left(\frac{v_b}{v_a}\right)^2 + \Delta c_W \qquad \textbf{(2.13.40)}$$

c_{WR} is now used to determine the vegetation roughness finally:

$$\lambda_{veg} = \frac{4 \cdot R_H \cdot d_p}{a_x \cdot a_y} + \Delta c_W \qquad \textbf{(2.13.41)}$$

The **total roughness** is the sum of **ground roughness** and **vegetation roughness**:

$$\lambda_{total} = \lambda_{veg} + \lambda_g \qquad \textbf{(2.13.42)}$$

which leads finally to the Manning roughness coefficient M as:

$$M = \sqrt{\frac{8 \cdot g}{\lambda_{total} \cdot R_H^{1/3}}} \qquad \textbf{(2.13.43)}$$

3. **Time step splitting:** Having calculated the flow velocities and the fluxes in m^3/s, the next step will be the calculation of the dynamic time step. Since all fluxes should flow to the neighboring cell only, the sub time step must be as short as required to allow the fastest flux to flow to the next cells border only.

$$t_{step} = \min_{i=1}^{n}\left(\frac{c_s \cdot c_{corr}}{v_i}\right) \quad \text{with} \quad \begin{array}{l} c_{corr} = 1.0 \text{ for flow direction 1, 3, 5 and 7} \\ c_{corr} = \sqrt{(2)} \text{ for flow direction 2, 4, 6 and 8} \end{array} \qquad \textbf{(2.13.44)}$$

with t_{step} sub time step in seconds
n number of active cells in the grids

c_s cell size [m]
c_{corr} correction factor accounting for orthogonal or diagonal directions
v_i flow velocity in cell *I*

WaSiM provides the possibility to limit both, the upper and the lower limit of the sub time step t_{step}. The duration of a time step is the absolute maximum for t_{step}, but the control file contains two parameters for the allowed range of the sub time step, one for the minimum sub time step ($t_{step,min}$) and one for the longest allowed sub time step ($t_{step,max}$). When setting $t_{step,min}$ to a significant larger value than the shortest flow time would suggest, the result will NOT be a complete unpredictable runoff concentration but rather a more retarded one – the fastest fluxes will only reach the next cell with a longer sub time step (and not the cell after next or even further). The limitation of $t_{step,max}$, however, to some smaller value than the fastest flow time suggests, may have positive effects: Since WaSiM uses a kinematic wave approach stating stationary conditions, backwater effects could only be accounted for when using smaller sub time steps. Thus, a very small gradient will not lead to oscillations in the flow directions.

4. The next step – the balancing of the fluxes – is optimized for parallel computing. Since WaSiM uses multiple flow paths and a cell may receive water from multiple other cells as well as it may send water to multiple other cells, the balancing is separated into two subsequent steps, each of which is parallelized. Otherwise, the adding of water from outflow into another cell could break the thread safeness (when two threads would add water into the same third cell, the addition of at least one flux would use the wrong initial value (i.e. the same initial value than the other thread is just using) and it would overwrite the result of the other thread, so water would be virtually lost.

   1. Calculating the totals of inflowing and outflowing water masses and tracer masses

$$q_{in,total} = \sum_{i=1}^{8} \hat{q}_{out,(i+3)\,mod\,8+1} \qquad \textbf{(2.13.45)}$$

$$q_{out,total} = \sum_{i=1}^{8} q_{out,i} \qquad \textbf{(2.13.46)}$$

   with $q_{in,total}$ total inflow into the actual cell from neighboring cells [m ^{3} /s]
   $\hat{q}_{out,(i+3)\,mod\,8+1}$ outflow of the cell laying in direction *i* into the actual cell-the flow direction for that cell will be calculated by (*i* +3) *mod* 8 +1, so e.g. the upper right cell with index 2 for the actual cell will contribute with its outflow in direction ((2+3) mod 8)+1 = 6, the upper left cell with index 8 with regard to the actual cell will contribute with its outflow in direction ((8+3) mod 8)+1 = 4 etc
   $q_{out,total}$ total outflow from the actual cell
   $q_{out,i}$ outflow from the actual into neighboring cell in direction *i*

   2. Balancing the actual water storage content for each cell:

$$m_{ss,t+t_{step}} = m_{ss,t} + 1000 \cdot \frac{q_{in,total} + q_{out,total}}{c_s^2} \cdot \Delta t \qquad \textbf{(2.13.47)}$$

   with m_{ss} amount of water in the surface routing storage [mm or kg/m^2] at time *t*
   Δt time step [s]

After balancing the water fluxes, the tracer masses are balanced using the actual fluxes and

the known tracer concentrations in the surface runoff storage.

5. For cells marked with a valid value in the river depth- and river-width grids, the value of m_{ss} (which is usually equivalent to $q_{in,total}$) will be added to the direct runoff grid (*QD*). For all other cells, the storage content in m_{ss} is then taken as $Q_{in,total}$ for the next sub time step
6. The sub time step counter is incremented and the scheme described in equations (2.13.17) to (2.13.29) is processed again (except calculation of the sub time step duration in eq. (2.13.26)) until the entire time step is finished.
7. To optimize the performance, the flow velocity calculation is skipped when all newly calculated weights (in any of the 8 possible directions) did not change by more than 1%. Then, the old velocities are used, again. However, the first flow velocity calculation in a time step will be carried out in any case, only subsequent *sub time steps* of the same time step may be optimized this way.
8. At the end of a time step, the total amounts of surface runoff and interflow are passed with its cell values into the single linear storage cascade to consider the translation and retention effect of the river network. To switch off this feature, simply no flow time grid has to be provided in the control file as standard grid.
9. The outputs from the single linear storage cascades for surface runoff (*QD*), interflow (*QI*) and base flow (*QB*) (with *QI* and *QB* taken directly from the unsaturated zone model) are then superposed to the total runoff (*QGES*) for each subbasin. This is the input for the routing model.
10. The content of the surface routing storage which did not ran off into a river cell, will be taken as input to the silting-up module or, if no silting-up module is used, to the unsaturated zone module in the next time step.

2.13.6 Control file extensions for Surface routing

Starting with WaSiM 8.01.01, a new submodel parameter set named `[SurfaceRoutingModel]` is added to the control file, which must be present in any control file (if surface routing is not used: at least the name of the submodel and the switch-parameter set to 0 must be present).

The example below shows a possible parameterization. The grid containing the surface water storage is shared between the two submodels “silting-up” and “surface routing”. It is created internally. To be able to write this grid to disk (which may be a good idea, especially when coupling different model runs, since the surface storage must be read in at the start of the coupled model run), the surface storage grid should be defined as variable grid with a default value of 0 (and a replace-nodata-value of 1).

The meaning of the parameters is described in the following example. There are two stacks, one for surface flow velocities and one for the amount of water flowing into each direction. These stacks are organized by flow direction: Level 1 contains the fluxes and velocities, resp., which are valid for direction 1 (North). Level 2 contains the fluxes and velocities valid for flow direction 2 (North-East) etc The 0-level contains the flux and velocity (in the respective stacks) in the main direction only.

The actual water level of all cells above see level (value of DHM + surface storage) is stored in the 9th layer of the surface velocity stack.

The single linear reservoir storage coefficients for surface runoff will be used from the parameterization of the unsatzon-model.

```
[SurfaceRoutingModel]
1                                   # 0=ignore this module, 1 = run the module
$time                               # duration of a time step in minutes
3                                   # method: 1=MultipleFlowPaths for diverging areas, 2=single
 (no line feed here in reality)          (nearest direction as given by aspect), 3 = Multiple Flow Paths
 (no line feed here in reality)          regarding water level) 4 = single flow paths (regarding water
level)
$outpath//qdsr//$grid//.//$year $hour_mean # direct discharge from surface routing module
$outpath//qisr//$grid//.//$year $hour_mean # interflow from surface routing module
$outpath//qbsr//$grid//.//$year $hour_mean # baseflow from surface routing module
$outpath//qgsr//$grid//.//$year $hour_mean # total discharge from surface routing module
$outpath//$surfspeed_grid           # stack with actual flow velocity of surface flow in m/s,the
0-element
 (no line feed here in reality)          contains the speed in main direction (or single direction flow)
$Writegrid                          # writegrid for this grid
$outpath//$surfflux_grid            # stack with actual flow amounts of surface flow in m^3/s, the
0-
 (no line feed here in reality)          element contains the flow in main direction
$Writegrid                          # writegrid for this grid
0.001                               # maximum wake lenght iteration difference (if Delta_A_nl <
this
 (no line feed here in reality)          value, iteration for a_NL stops)
40                                  # maximum number of iterations for a_NL
0.0001                              # maximum flow velocity iteration difference (if Delta v is
less than
 (no line feed here in reality)          this value, iteration stops)
40                                  # maximum number of iterations for v
30                                  # shortest sub-time step in seconds
300                              #longest allowed sub time step (even if flow travel times are longer,
 (no line feed here in reality)       the time step is subdivided into sub timesteps of this lenght) be
 (no line feed here in reality)       careful: tracers are mixed much faster when multiple sub time steps
 (no line feed here in reality)       are applied
0.04                                    # minimum water depth for regarding roughenss of crops in m
(shallower
 (no line feed here in reality)          sheet flow: only roughness of bare soil will be regarded)
3.0                              # ConcentrationFactor takes into account the micro scale concentration
 (no line feed here in reality)          of flow pathes, flow will take place on a fraction of the cell
only,
 (no line feed here in reality)          so the amount flowing per meter width will be multiplied by this
 (no line feed here in reality)          factor (1..n)
$readgrids                       # readgrid code 0 do not read, 1 = read grids
```

Furthermore, some land use specific parameters are used in the land use table. Here is an example with parameter description. The additional parameters for surface routing are printed in blue/bold font:

```
6 agriculture     {method    = VariableDayCount;
    SoilTillage      = 90 240;# optional set of 1..n Julian days, depicting days with soil
                              # tillage. Important for silting up model
   RootDistr        = 1.0;    # parameter for root density distribution
   TReduWet         = 0.95;   # relative Theta value for beginning water stress (under wet
                              #  conditions -> set >= 1 for crop which doesn't depend on an aeral
                              #  zone
   LimitReduWet     = 0.5;    # minimum relative reduction factor of real transpiration when water
                              # content reaches saturation. The reduction factor value will go down
                              # linearly starting at 1.0 when relative Theta equals TReduWet (e.g.
                              # 0.95) to LimitReduWet when the soil is saturated (Theta rel = 1.0)
   HReduDry         = 3.45;   # hydraulic head (suction) for beginning dryness stress (for water
                              # content resulting in higher suctions, ETR will be reduced down to 0
                              # at suction=150m)
   IntercepCap      = 0.3;    # optional: specific thickness of the water layer on the leafes in mm.
                              # if omitted here, the dedfault parameter from interception_model is
                              # used
   StartVegetationPeriodForBalance = 3 ; # the sampling point in the following JD-Table when the
                                 # vegetation period starts, default = 0 (start of model run)
   StopVegetationPeriodForBalance = 10 ; # the sampling point in the following JD-Table when the
                                    vegetation period ends, default = n+1 (end of model run)
   JDVegetationResetForBalance = 1 ; # Julian day, when vegetetaion start and vegetation stop grids
                                     are re-initialized to -1 (northern hemisphere: usually day
1)
   JulDays          = 15    46    74    105   135   166    196    227    258    288    319    349 ;
   Albedo           = 0.25  0.25  0.25  0.25  0.25  0.25   0.25   0.25   0.25   0.25   0.25   0.25;
   rsc              = 80    80    75    75    65    65     65     65     65     75     90     90;
   rs_interception  = 80    80    75    75    65    65     65     65     65     75     90     90;
```

```
   rs_evaporation    = 240  240   240   240   240    240    240   240    240    240    240    240;
   LAI               = 1     1     2     3     4     5      5     4      3      2      1      1;
   Z0                = 0.05  0.05  0.1   0.25  0.5   0.5    0.5   0.4    0.3    0.15   0.05   0.05;
   VCF               = 0.3   0.3   0.3   0.5   0.8   0.8    0.8   0.7    0.6    0.3    0.3    0.3 ;
   RootDepth         = 0.05  0.05  0.1   0.2   0.3   0.4    0.4   0.4    0.3    0.2    0.05   0.05;
   AltDep          = 0.025 0.025 0.025 0.025 0.025 0.025 -0.025 -0.025 -0.025 -0.025 -0.025 -0.025;
# now some parameters for surface routing will follow. They are optional, the default value is
# given at the start of the comment.
   SR_cultivation    = nonconserving; # default: other; type of cultivating the surface; possible
                                      # alues: conserving (c), nonconserving (nc), other (o),
abbrev.
                                      #used in the following rows are for convenience only
(nc) SR_StemDiameter  = 0.01  0.02  0.03 0.04 0.05 0.05  0.08  0.08  0.05  0.03  0.02  0.02;
              # default: 0.01 surface routing: diameter of the stalks or effective diameter of
              # crop/tree stems (perpendicular to the streamlines); used for nonconserving land use
              # (using ploughs)
(c)  SR_ResidCovrg    = 0.3   0.4   0.6  0.7  0.8  0.9   0.9   0.9   0.6   0.5   0.4   0.3;
                              # default: 0.01 fraction of soil covered by vegetation, stones, and
                              # residuals (mulch); used for conserving land use only
(nc) SR_CropDistX     = 0.3;     # default: 0.3 distance of crops in flow direction; used for
                              # nonconserving land use (using ploughs) only
(nc) SR_CropDistY     = 0.4;     # default: 0.3 distance of crops perpendicular to flow direction;
                              # used for nonconserving land use (using ploughs) only
 (nc/o) SR_K_ST        = 10;      # default: 10 Manning-Strickler-roughness for the soil surface in
                              # (m^1/3)/s; used for nonconserving and other landuse only
(nc) SR_C_WR          = 1.5;     #default: 1.5 form drag coefficient for a circular cylinder in a
                              # group; used for nonconserving landuse only
(c)  SR_rough_coeff1  = 0.00496 0.892 0.0311; # default: 0.0198 0.712 0.142; coefficients for
                                               #  k_st-calculation for conserving landuse, Reynolds
                                               #  number < 20000
(c)  SR_rough_coeff2  = 0.00519 1.200 0.177 ; # default: 0.0198 0.712 0.142; coefficients for
                                               #  k_st-calculation for conserving landuse, Reynolds
                                               #  number >= 20000
              }
```

2.13.7 Extension to surface routing module with 2D equation system

Starting with version 10.00.00 (or patch 9.10.02), a new solution algorithm for surfacxe routing is available. This algorithm uses a 2D flow equation that is solved in a system of linear equations. The efficiency is much better than for the traditional local approach (much less iterations required or longer sub time steps possible) and the stability of the local surface e.g. on lakes is much better. However, a drawback is, that it is still a 2D approach. That means that 3D features like (small) dams or large differences in flow depths between cells can lead to artifacts like water flowing uphill over dams when the level of accuracy is not chosen high enough.

The key features of this 2D-approach are

- very similar approach to 2D groundwater model
- flow equations are solved in a system of linear equations by using the Gauss-Seidel-algorithm (unlike the Manning-Strickler equation, where slope, hydraulic radius and roughness are non-linear factors of the solution)
- optimized for parallel computing
- fast solution due to the implicit finite difference algorith (Gauss-Seidel uses a moving 3x3 window that uses heads for time step n+1 for two of the four directions in every time step, so the algorithm is not completely implicit, but partly, which is good enough here as long as at least two iterations are done)
- 4 flow directions are possible for each cell, depending on the local slopes (a special flow direction algorithm for generating flow channels is available in Tanalys to make sure that channels are not connected diagonally via the corners only but always via the edges of the cells →)
- no differences in number of flow paths between converging and diverging areas
- no concentration factor can be used
- no distinction between arable land use and natural areas, there is only one roughness parameters used that is part of the land use table (optional parameter SR_K_ST)
- tracers will not be modeles with this approach

The new algorithm solves the flow equation for all cells simultaneously in a big equation system by using an iterative solver (Gauss-Seidel). The algorithm itself is a finite difference algorithm. The basic flow equation is very similar to the groundwater flow equation. That means that there is no non-linear but rather a linear dependency between the flow velocity and the surface roughness. The flow velocity depends on the slope and the local transmissivity, which is expressed as the product of a roughness coefficient and the water depth. This is the main mathematical difference between the old method, using the Manning-Strickler flow equation with multiple non-linear dependencies .

The old method uses equation (2.13.48)

$$Q = M \cdot A \cdot R^{2/3} \cdot S^{1/2} \qquad \textbf{(2.13.48)}$$

where Q is the flow in m ^{3} /s, M is the Manning coefficient (a roughness factor) in m $^{1/3}$ /s, A is the area of the effective flow profile in m ^{2} , R is the hydraulic radius in m and S is the slope in m/m.

The old approach solves equation (1) directly for the flow rate for each flow path locally, which is the bid advantage of this approach (flow velocity and flux amounts are direct results). The big disadvantage is the instability, requiring extremely short time steps and many iterations. And even with time steps of a few seconds, the equations were not stable enough for lake surfaces.

The new algorithm uses a system of linear equations by applying a finite difference method. The flows are not explicitly calculated but only implicit by computing new pressure heads for given time steps.

T he basic equation is

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = \frac{S}{T} \cdot \frac{\partial h}{\partial t} - \frac{R(x,y,t)}{T} \qquad \textbf{(2. 13.49)}$$

where the left side describes the pressure head change in x and y direction (2D flow) caused by different inflows and outflows (i.e. the difference between inflow and outflow into a specific volume) and the left side describes the change in storage volume by time, with S [-] as storage coefficient (1 for surface flow), T in m ^{2} /s as transmissivity (equals roughness times water layer thickness) and R in m/s as time dependent recharge.

E quation (2) is identical to the groundwater flow equation but applied to surface water flow here. The hydraulic conductivity is replaced by the surface roughness. The resulting transmissivity is used like in the groundwater model. The algorithm is applied iteratively using the Gauss-Seidel solution - which can be parallelised very efficiently for super computers even when using MPI parallelism when doing at least three or four iterations for each sub time step (in order to exchange the results between the MPI ranks that are inputs to the rows of the next rank for Gauss-Seidel).

The big advantages of this method are:

i) the equation is stable enough to create smooth surfaces even on lakes, so it can handle even backwater consitions

ii) since the approach is stable and the convergence criterion is reached within only a few iterations, the approach is very fast compared to the old one (up to 10 times faster).

However, there is one big disadvantage: The solution algorithm calculates new pressure heads for each cell for the next time step – but no flow velocity and no flux amount directly. This is a general disadvantage of finite difference approaches (that could be overcome by using finite element approaches) . One could argue that i t should be a simple task to calculate those values by simply taking the head difference between two cells and multiplying this with the local transmissivity to get the flow velocity. But that won't work since the difference in pressure heads is changing during the time step and we do not know the exact flow velocity and hence we do also not know the absolute water flux (in m ^{3} /s). WaSiM solves this problem by keeping the total amount of all

surface water in balance with the outflow into river cells. If there is only one river cell defined for a sub basin, then this approach works fine – all local changes in head (i.e. change in water table on a cell) are summed up as total storage change and will then be set as the total flux within the last time step that flew into that river cell. If multiple river cells are defined, the total change in surface storage is divided up weighed by the local pressure head gradients for each single outflow, again with the sum of all local outflows resulting in the total storage change.

A nother disadvantage is the fact that the approach doesn't know about the real surface profile. It only knows about relative pressure heads (for neighboring cells) and transmissivities. That makes it possible that water may flow upwards locally if using too less iteration steps. Imagine three neighboring cells, where the first and the third cell have the same ground level and water table of 1 m and 1.1 m above ground. The cell in - between has a ground level that is 1.2 m above the ground level of the first and third cell. So this is like a small dam. No flux is possible from cell one to cell three – as long as there is now water on cell two, which then is internally a no-flow boundary condition. But as soon as there is some water on cell two , all three cells will be part of the equation system. Since the ground levels and therefore also the pressure heads needs to be adjusted to prevent the water level to drop below the cells ground and since different water depths will only be accounted for by the transmissivities, the algorithm tends to allow drainage of isolated (groups of) cells as long as there is a "watery" connection to other cells.

T he method to set in the control file for using this new methid is 6.

The maximum number of iterations and the maximum allowed error in can be set using the variables for the velocity iteration in the old approaches:

```
[SurfaceRoutingModel]
1                              # 0=ignore this module, 1 = run the module
$time                          # duration of a time step in minutes
6                              # method: recommended: new method 6=Gauss-Seidel
…
0.000001    # maximum flow velocity iteration difference
1000           # maximum number of iterations for v
```

To create a channel system using only four flow direction instead of eight, the control fiel for tanalys needs to be changed: the method for [Flowdirection] needs to be set to a value greater than 1:

```
[Flowdirection]
2                           # switch 0=do not calculate, 1=calculate this section
2=use 4-way flow direction for new surface routing model
$Path//$Flowdirection     # result grid corrected flow directions
$Path//$Grid//.sin        # result grid sinks
$Path//$Grid//.dhk        # result grid corrected DEM (filled sinks)
```

2.14 Soil model for the WaSiM version using the Richards-equation

2.14.1 Introduction

WaSiM versions with physically based soil model uses the RICHARDS-equation for modeling the fluxes within the unsaturated soil zone. The modeling is done one-dimensional in the vertical direction using a soil with several numeric layers. The continuity equation for this type of problem is given by:

$$\frac{\partial \Theta}{\partial t} = \frac{\partial q}{\partial z} = \frac{\partial}{\partial z}\left(-k(\Theta)\frac{\partial \Psi(\Theta)}{\partial z}\right) \qquad \textbf{(2.14.1)}$$

with Θ water content [m^3/m^3]
t time [s]
k hydraulic conductivity [m/s]
Ψ hydraulic head as sum of the suction ψ and geodetic altitude h [m]
q specific flux [m/s]
z vertical coordinate [m]

The discretised RICHARDS-equation is given by:

$$\frac{\Delta \Theta}{\Delta t} = \frac{\Delta q}{\Delta z} = q_{\text{in}} - q_{\text{out}} \qquad \textbf{(2.14.2)}$$

with q_{in} inflow into the actual soil layer [m/s]
q_{out} outflow from the actual soil layer (including interflow and artificial drainage) [m/s]

The dependencies of the hydraulic properties on the water content of the soil are considered in a discrete manner. The flux q between two layers with indices u (upper) and l (lower) is then given by:

$$q = k_{\text{eff}} \cdot \frac{h_h(\Theta_u) - h_h(\Theta_l)}{0.5 \cdot (d_u + d_l)} \qquad with \qquad \frac{1}{k_{eff}} = \frac{d_u}{d_l + d_u} \cdot \frac{1}{k\left(\Theta_u\right)} + \frac{d_l}{d_l + d_u} \cdot \frac{1}{k\left(\Theta_l\right)} \qquad \textbf{(2.14.3)}$$

with q flux between two discrete layers [m/s]
k_{eff} effective hydraulic conductivity [m/s]
h_h hydraulic head, dependent on the water content and given as difference of geodetic altitude h_{geo} [m] and suction $\psi(\Theta)$ after equation (2.14.4)
d thickness of the layers under consideration [m]

In equation (2.14.3) , the effective conductivity is the harmonic average of the conductivities of both layers. It is possible to use layers with any thickness for this approach.

2.14.2 Parameterization

The parameterization of the hydraulic properties of the soil can be setup by using two different methods. The first method uses look-up-tables for the relations between the hydraulic head h_h and Θ as well as between the hydraulic conductivity k and Θ, respectively. For each soil type, a free number of table entries [Θ, $h_h(\Theta)$] and [Θ, $k_{rel}(\Theta)$] can be given. k_{rel} is the relative hydraulic conductivity dependent on the saturation of the soil (0..1). The saturated hydraulic conductivity k_s can be given for each soil layer. The model interpolates between the table entries. The advantage of this method is the possibility to fit the parameters to any type of soil. However, a lot of observation data is required. Therefore, the second method uses common parameters by applying the method

after VAN GENUCHTEN (1976):

$$\psi(\Theta)=\frac{1}{\alpha}\left[\left(\frac{\Theta-\Theta_r}{\Theta_s-\Theta_r}\right)^{-1/m}-1\right]^{1/n} \qquad \textbf{(2.14.4)}$$

and

$$\frac{k(\Theta)}{k_s}=\left[\frac{\Theta-\Theta_r}{\Theta_s-\Theta_r}\right]^{1/2}\cdot\left[1-\left\{1-\left(\frac{\Theta-\Theta_r}{\Theta_s-\Theta_r}\right)^{1/m}\right\}^m\right]^2 \qquad \textbf{(2.14.5)}$$

with ψ suction [m]
α empirical parameter [m^{-1}]
n empirical parameter [-]
m empirical parameter [-] with $m = 1\text{-}1/n$
Θ actual water content [-]
Θ_r residual water content at $k(\Theta) = 0$ [-]
Θ_s saturation water content [-]
k_s saturated hydraulic conductivity [m/s]

Using equation (2.14.4) the actual water content is given by the actual suction by:

$$\Theta=\Theta_r+\left(\Theta_s-\Theta_r\right)\left(\frac{1}{1+(\psi\alpha)^n}\right)^m \qquad \textbf{(2.14.6)}$$

Besides the parameters n and α a third parameter k_{rec} has to be specified for each soil type for describing the recession of the saturated conductivity with depth:

$$k_{s,z}=k_s\cdot k_{rec}{}^z \qquad \textbf{(2.14.7)}$$

with $k_{s,z}$ saturated hydraulic conductivity within depth z [m/s]
k_s saturated hydraulic conductivity at the soil surface [m/s]
k_{rec} recession constant [-]
z depth [m]

2.14.3 Numerical solution (vertical fluxes, interflow, infiltration/exfiltration in rivers)

For solving equation (2.14.3) a one dimensional vertical finite difference (FD) scheme is applied. After transforming equation (2.14.1) into its discrete form as given by equation (2.14.2) the fluxes between the discrete soil layers are calculated followed by the calculation of interflow, drainage, and exfiltration into or infiltration from rivers. The upper boundary condition is the infiltration amount F which is the minimum of either the fillable porosity of the uppermost numerical soil layer and the precipitation amount per time step (or melt water etc.). The FD-scheme starts in the lowest layer which was completely saturated by the infiltration or, if no layer was saturated, in the uppermost layer. The lower boundary condition is the depth of the groundwater layer, which is constant for a specific time step but which is variable in time (due to the groundwater flow as well as due to recharge or capillary rise). The extraction of evaporating water is done before the FD-scheme starts (see chapter 1.12.4). The FD-scheme is done in the following steps 1 to 7, where the points 2 to 7 are iteratively repeated until the scheme is stable, which is determined by the change in water content from step to step or until a maximum number of iteration steps is reached (usually not more than 100 … 1000).

1. *Calculating infiltration* down to layer *m*, it's possible to generate interflow at layer boundaries

2. *flux calculation:* It starts from layer *m-1,* the last completely saturated layer (or from layer 1, if $m = 1$) down to the groundwater layer. The flux q_m (flux between layers m and $m+1$) is estimated after equation (2.14.3) . The actual water content of the layers m and $m+1$ are estimated by:

$$\begin{aligned}\Theta_m &= 0.5\left(\Theta_{m,i-1} + \overline{\Theta_{m,iter}}\right)\\ \Theta_{m+1} &= 0.5\left(\Theta_{m+1,i-1} + \overline{\Theta_{m+1,iter}}\right)\end{aligned} \qquad \textbf{(2.14.8)}$$

with Θ water content [-]
m index of the layer (in ascending order from top of soil downwards)
i time step
$\overline{\Theta_{m,iter}}$ accumulated average of the water content from all iteration steps, beginning with iteration step 4

For reasons of convergence and stability, especially if using large time steps, after the 4th time step the water content is calculated as an average using the value from the last time step and the accumulated average from all iteration steps in the actual time step (starting with iteration step 4). For the first iteration step, Θ_m and Θ_{m+1} are taken from the results of the last time step. For iteration steps 2 and 3, the Θ -values are estimated as an average from the last time step $\Theta_{m,i-1}$ and the result from the last iteration step $\Theta_{m,i,iter}$.

Using these water content values, the hydraulic conductivities k, the suction values ψ, and the hydraulic heads h_h are estimated for upper and lower layer using equations (2.14.4) and (2.14.5) . It holds $h_{h,m} = \psi_m + h_{geo,m}$ (see equation 82). Because of the discretization in space and time there is a constraint for the relation of the conductivity values of two adjacent layers: Both, the value of $k(\Theta_m)/k(\Theta_{m+1})$ and $k(\Theta_{m+1})/k(\Theta_m)$ has to stay below a maximum value in order to avoid an artificial "flow obstacle" which would be there if, for instance, the layer m is saturated but layer m+1 is still near by or at wilting point:

$$\left.\begin{aligned}k_m &= k(\Theta_m)\\ k_{m+1} &= k(\Theta_{m+1})\end{aligned}\right\} \text{ limited to } \left\{\begin{aligned}k_m &= \frac{1}{c_k}k(\Theta_{m+1})\\ k_{m+1} &= \frac{1}{c_k}k(\Theta_m)\end{aligned}\right\} \text{ if } \left\{\begin{aligned}k_m &< \frac{1}{c_k}k(\Theta_{m+1})\\ k_{m+1} &< \frac{1}{c_k}k(\Theta_m)\end{aligned}\right. \qquad \textbf{(2.14.9)}$$

with c_k empirical parameter, dependent on the vertical discretization and on the time step ($c_k \approx 50 \ldots 100$ at $1\text{h} \le \Delta t \le 5\text{d}$ and $\Delta z \approx 0.33$ m)

3. *Calculating interflow:* Interflow is generated if the suction ψ_m in a layer m holds: $\psi_m < 3.45$ m. The calculation is performed in two steps, in a third step the comparison between the results is done:

a) maximum possible interflow $q_{ifl,max}$ as drainable water content of the actual layer

$$q_{ifl,max} = (\Theta(\psi) - \Theta_{\psi=3.45}) \cdot \frac{\Delta z}{\Delta t} \qquad \textbf{(2.14.10)}$$

with $\Theta(\psi)$ water content at actual suction ψ [-]
$\Theta_{\psi=3.45}$ water content at suction $\psi = 3.45$ m [-]
Δz layer thickness [m]
Δt time step [s]

b) Interflow q_{ifl} as a result of conductivity, river density and hydraulic gradient:

$$q_{\text{ifl}} = k_s(\Theta_m) \cdot \Delta z \cdot d_r \cdot \tan\beta \qquad \textbf{(2.14.11)}$$

with k_s saturated hydraulic conductivity [m/s]
Θ_m water content in the actual layer m [-]
d_r scaling parameter used to consider river density as well as effects like a anisotropy of $k_{s,horizontal}$ compared to $k_{s,vertical}$ [m^{-1}]
β local slope angle (**important:** if $\beta > 45°$, β is limited to 45°)

c) The result is the smaller value of steps a) and b).

4. *Infiltration from rivers and exfiltration from groundwater into rivers*:

If a grid cell is marked as a river cell (which is done by having valid entries in the river width grid, the river depth grid, and the leakage factor grid), exfiltration into the river bed can be calculated. Infiltration from rivers into groundwater can be calculated only if there is in addition an entry in a so called link grid, a grid containing the ID-number of the tributary which is sending it's water downstream to the actual cell. Thus, infiltration into groundwater is possible only at grid cells which are occupied by real routing channels, whereas exfiltration is possible at each grid cell which is marked as river. This scheme is based on the assumption that in smaller headwater subbasins usually the gradient of flow if directed from the groundwater to the rivers. Only in large stream valleys (which are usually routing channels) this gradient can be reversed if there are large amounts of external inflows from the upper areas.

The estimation of river width and river depth as well as the marking of routing channels is done in the preprocessing using the program TANALYS. It is also possible to use observed values by generating grids from any GIS-Coverage or from other bases.

The equations for estimating exfiltration and infiltration are relatively simple:

Exfiltration (which is the base flow), calculation has two steps:

a) calculation of exfiltration using the hydraulic gradient and the colmation (in- and exfiltration resistance) at the river bed:

$$q_{\text{exf,pot}} = l_k \cdot \Delta H \cdot b_{\text{rb}} / cs \qquad \textbf{(2.14.12)}$$

with $q_{exf,pot}$ maximum possible exfiltration (base flow) [m/s]
l_k Leakage-factor (colmation resistance) [s^{-1}]
ΔH positive difference between groundwater table and river bed
$\Delta H = h_{GW} - h_{rb}$ (h_{GW}: groundwater table [m a.s.l.], h_{rb}: river bed [m a.s.l.])
b_{rb} width of the river bed [m]
cs grid cell size [m]

b) partitioning of $q_{exf,pot}$ to all affected soil layers: From each layer water can be extracted only until the pre-defined suction of 3.45 m is reached:

$$q_{exf,m} = \left(\Theta_m - \Theta_{\psi=3.45\text{m}}\right) \cdot \Delta z_e / \Delta t$$

$$\text{for} \quad \text{int}\left((h_{\text{geo},0} - h_{GW})/\Delta z\right) \le m \le \text{int}\left((h_{geo,0} - h_{\text{rb}})/\Delta z\right) \qquad \textbf{(2.14.13)}$$

with $q_{exf,m}$ maximum possible exfiltration from layer m [m/s]
m layer index, starting from the uppermost layer with contact to the groundwater down to the layer the river bed is within [-]
Θ_m water content in layer m (usually saturation) [-]
$\Theta_{\psi=3.45\text{m}}$ water content at suction ψ = 3.45 m [-]
$h_{geo,0}$ surface altitude [m a.s.l.]
h_{GW} groundwater head [m a.s.l.]

h_{rb} altitude of the river bed [m a.s.l.]
Δz layer thickness [m]
Δz_e effective layer thickness: If the layer is positioned completely between groundwater surface and river bed, then $\Delta z_e = \Delta z$ (overall layer thickness); if the groundwater table is positioned within layer *m*, then $\Delta z_e = h_{GW} - h_{geo,m}$; if the river bed is positioned within layer *m*, then $\Delta z_e = h_{geo,m-1} - h_{rb}$; if both, the groundwater table and the river bed are positioned within layer *m*, then $\Delta z_e = h_{GW} - h_{rb} = \Delta H$

If for the uppermost layer *m* the amount $q_{exf,m}$ is smaller than $q_{exf,pot}$, the remaining amount is taken from the next layer (or at least that part which can be taken from the next layer) and so on - as long as this next layers location is at least fractionally above the river bed. On the other side, if $q_{exf,pot}$ is smaller than $q_{exf,m}$, the amount taken from layer *m* is limited to $q_{exf,pot}$. The amounts taken from each layer are subsumed:

$$q_{\mathrm{exf}} = \sum_m q_{exf,m} \qquad \textbf{(2.14.14)}$$

with q_{exf} amount of exfiltrated water

Infiltration: Like the estimation of exfiltration also the estimation of infiltration is done using equation (2.14.12) . There are also two steps:

a) potential infiltration as given by the hydraulic gradient and by the transition resistance (colmation):

$$q_{\mathrm{inf},pot} = l_k \cdot \Delta H \cdot b_{\mathrm{rb}} / cs \qquad \textbf{(2.14.15)}$$

with $q_{inf,pot}$ maximum possible infiltration as given by the gradient [m/s]
l_k leakage-factor (transition resistance or colmation resistance) [s^{-1}]
ΔH positive difference between river bed altitude and groundwater table $\Delta H = h_{rb} - h_{GW}$; if the groundwater table is below the river bed it holds: $\Delta H = h_{rb} - hh_n$, whereas hh_n is the hydraulic head of the layer the river bed is positioned within
b_{rb} width of the river bed [m]
cs grid cells size [m]

b) In analogy to equation (2.14.13) , the potential infiltration is filled into the soil layers starting with the layer the river bed is located in down to the groundwater table. The fillable porosity is the difference between the saturation water content and the actual water content as valid in the actual iteration step. Darcy's law is not necessarily considered, this means that there may be more infiltration than the hydraulic conductivity would allow. To limit the infiltration amount, the leakage factor should be reduced.

$$q_{\mathrm{inf},m} = (\Theta_{sat} - \overline{\Theta_{m,iter}})\Delta z / \Delta t \qquad \textbf{(2.14.16)}$$

with $q_{inf,m}$ maximum infiltration into the layer *m* as given by the fillable porosity [m/s]
$\overline{\Theta_{m,iter}}$ actual water content in the layer *m* as taken from the actual iteration step *iter* [-]

Like for exfiltration, also for infiltration a sub-summation is done during the filling of the layers:

$$q_{\mathrm{inf}} = \sum_m q_{\mathrm{inf},m} \qquad \textbf{(2.14.17)}$$

with q_{inf} infiltration amount which is already filled into empty layers [m/s]

If after the filling of all layers below the river bed this sum q_{inf} is still less than the potential

infiltration $q_{inf,pot}$, then the layers in-between the river bed and the water table in the river are filled starting from the river bed up to the water table. If afterward the potential infiltration is still not completely filled into the soil, the real infiltration q_{inf} is limited to the actual accumulated value $\Sigma q_{inf,m}$. The total infiltration amount (in units of m^3) is stored in a table with entries for each routing channel. This table contains also the discharge within each routing channel as calculated by the routing-module (results from the previous time step). After calculating the infiltration the infiltration amount for the actual cell is subtracted from the discharge amount of the matching channel in the discharge table. If the infiltration after equations (2.14.15) to (2.14.17) is larger than the available amount of water in the channel, the infiltration is limited to that available amount in the channel. The channel is drying out.

5. g*radient restrictions*: Because of the discretization in time and space (in WaSiM the time step is constant and usually equal or greater than one hour) some restrictions for the fluxes must be applied. For instance if there is a very large flux from layer *m* into layer *m*+1 over a long time step which leads to a considerable decreasing water content in layer *m* and to an increasing water content in layer *m*+1, the hydraulic gradient may switch from downwards to upwards within this time step. In order to keep the numbers of iteration steps low, for each calculated flux the resulting water contents are checked and adjusted if required. The first step is a check for over-saturation (later in point 6 further flux restrictions are applied). The balance of inflows and outflows with the water content of the actual layer *m* in the time step *i* is:

$$\Theta_{m,i+1} = \Theta_{m,i} + (q_{m,i} - q_{ifl,i} - q_{m-1,i} - q_{\mathrm{exf},m} + q_{\mathrm{inf},m}) \frac{\Delta t}{d_m} \qquad \textbf{(2.14.18)}$$

with $\Theta_{m,i}$ $\Theta_{m,i+1}$ Water content before and after the actualization with the calculated fluxes
$q_{m,i}$ flux between layers *m* and *m*+1 (outflow has a negative sign) [m/s]
$q_{ifl,i}$ interflow from layer *m* [m/s]
$q_{m-1,i}$ flux between layers *m*-1 and *m* (inflow has a negative sign) [m/s]
$q_{exf,m}$ exfiltration from the ground water, calculated after equation (2.14.12) /(2.14.13) [m/s]
$q_{inf,m}$ infiltration into the groundwater, calculated after equation (2.14.14) /(2.14.15) [m/s]
Δt time step [s]
d_m thickness of layer *m* [m]

If the balance is greater than the saturation water content, the flow from layer *m*+1 into layer *m* is reduced. The inflow from above (from layer *m*-1) is automatically reduced afterward when the flux restrictions are applied to the layers above (the flux restriction is done from the lower to the upper layers of the soil column).

Also for layer *m*+1 a balance is calculated after equation (2.14.18) with the resulting water content $\Theta_{m+1,i+1}$. Using both water content values, the corresponding hydraulic heads $hh_{m,i+1}$ and $hh_{m+1,i+1}$ are calculated:

$$\begin{aligned} hh_{m,i+1} &= 0.5(h_{geo,m-1} + h_{geo,m}) + \psi(\Theta_{m,i+1}) \\ hh_{m+1,i+1} &= 0.5(h_{geo,m} + h_{geo,m+1}) + \psi(\Theta_{m+1,i+1}) \end{aligned} \qquad \textbf{(2.14.19)}$$

with hh hydraulic head [m]
h_{geo} altitude of the lower boundary of the layer [m a.s.l]
ψ suction [m]
i index for time
m index for layers

A gradient restriction is applied if equation (2.14.20) holds:

$$(hh_{m,i} > hh_{m+1,i}) \wedge (hh_{m,i+1} < hh_{m+1,i+1}) \vee (hh_{m,i} < hh_{m+1,i}) \wedge (hh_{m,i+1} > hh_{m+1,i+1}) \quad \textbf{(2.14.20)}$$

In such cases the time t_s at which the gradient becomes 0 is estimated. To do so, the balances after equation (2.14.18) are calculated for the middle of the time step using $\Delta t_{neu} = ½ \Delta t$ as Δt. The new water content values are then $\Theta_{m,i+½}$ and $\Theta_{m+1,i+½}$. After equation (2.14.19) the new hydraulic heads are $hh_{m,i+½}$ and $hh_{m+1,i+½}$. These heads are compared to the heads from the beginning of the time step. If the gradients are still reversed, the time step is divided again into two halves. If the gradient is not reversed but larger than a certain threshold (of e.g. 0.001m), the time step is enlarged by 50 % of the previously cut time slice. Again, the iterative search for the time of gradient reversal is continued. If this time t_s is found, the fraction of this time on the time step duration is the reduction factor for the calculated fluxes:

$$q_{m,red} = \frac{t_s}{\Delta t} \cdot q_m \quad \textbf{(2.14.21)}$$

with $q_{m,red}$ reduced flux between 2 layers, in order to avoid gradient reversals [m/s]
t_s time from the beginning of the time step until the gradient is zero [s]

6. *flux restrictions*: The gradient restriction is applied each time only for two adjacent layers, even if the fluxes out of or into the next layers are considered for the balances, they still can be changed due to the following gradient restriction of the next layer. Therefore the gradient restriction is done a few times in an iterative scheme. In addition after this scheme it is checked if there are cases of "over-saturation" or "under-saturation" (drying, $\Theta < \Theta_w$). Because of the downwards decreasing dynamic of the fluxes in the unsaturated zone the smallest corrections should be at the lower boundary. Also fluxes are usually restricted only from upper to lower layers, what may lead to water logging in the upper layers. Thus it is recommended to perform the flux restriction starting from the lowest unsaturated layer up to the soil surface. There are two possible cases:

6.1 $\Theta_{m,i+1} > \Theta_s$ (balance after equation (2.14.18) : over saturation, reducing inflows)

a) q_m and q_{m+1} are flowing downwards (vertical throughflow); reducing q_{m-1} and q_{inf}

$$q_{m-1,red} = q_{m-1} + (\Theta_{m,i+1} - \Theta_s) \cdot \Delta z / \Delta t \quad \textbf{(2.14.22a)}$$

with $q_{m-1,red}$ reduced vertical inflow from the upper layer
q_{m-1} original value of the inflow from the upper layer
Δz layer thickness [m]
Δt time step [s]
$q_{inf,m}$ infiltration after equation (2.14.16)
Θ_s saturation water content [-]
$\Theta_{m,i+1}$ water content as balance after equation (2.14.18) from the unrestricted fluxes [-]

If the reduction after (2.14.22a) should lead to an inverse sign of $q_{m-1,red}$ compared to q_{m-1}, the reduction was to large. The reduction is then split into a fraction which reduces q_{m-1} down to zero and another fraction which is applied to the second possible inflow, to the infiltration from surface water:

$$q_{ifl,m,red} = q_{ifl,m} - q_{m-1,red} \quad \text{and} \quad q_{m-1,red} = 0 \quad \textbf{(2.14.22b)}$$

with $q_{inf,m,red}$ reduced infiltration from a surface water (river) [m/s]
$q_{inf,m}$ infiltration after equation (2.14.16)

b) inflows from below and from above (q_m:↑, q_{m-1}:↓, additional inflow possible by $q_{inf,m}$)
In a first step the inflow from above is reduced after equation (2.14.22a) :

$$q_{m-1,red} = q_{m-1} + (\Theta_{m,i+1} - \Theta_s) \cdot \Delta z / \Delta t \qquad \textit{(identical with (2.14.22a))}$$

If the sign of the inflow from above should be inverted by the reducing, the reduction itself is reduced in a way that the inflow from above is reduced to zero. The inflow from below is reduced by (note that the reduction of the inflow from above is corrected after the reduction of the inflow from below! The sign of $q_{m-1,red}$ is now positive after the over-reduction!):

$$q_{m,red} = q_m - q_{m-1,red} \quad \text{and} \quad q_{m-1,red} = 0 \qquad \textbf{(2.14.22c)}$$

If this reduction of the inflow from below also leads to a switch of the sign of the flux $q_{m,red}$ compared to q_m, the reduction is again reduced in a way that the flux from below is set to zero. The remaining amount of reduction is then applied to the infiltration from surface water. Note that in equation (2.14.22d)the correction term is negative, because during the previous reduction of q_m to $q_{m,red}$ the sign of the flux from below has changed from + to -.

$$q_{ifl,m,red} = q_{ifl,m} + q_{m,red} \quad \text{and} \quad q_{m,red} = 0 \qquad \textbf{(2.14.22d)}$$

c) only inflow from below and from infiltration, other fluxes are outflows or zero (q_m ↑, q_{m-1} ↑, $q_{inf,m} \geq 0$). In a first step the inflow from below is reduced:

$$q_{m,red} = q_m - (\Theta_{m,i+1} - \Theta_s) \cdot \Delta z / \Delta t \qquad \textbf{(2.14.22e)}$$

Again, if the sign of the flux q_m is changed, the reduction itself has to be reduced and the remaining reduction has to be applied to the infiltration from surface water:

$$q_{ifl,m,red} = q_{ifl,m} + q_{m,red} \quad \text{und} \quad q_{m,red} = 0 \qquad \text{(identical with (2.14.22d))}$$

c) only inflow from infiltration from rivers, all other fluxes are outflows (q_m ↓, q_{m-1} ↑, $q_{inf,m} \geq 0$). Only the infiltration from rivers is reduced:

$$q_{ifl,m,red} = q_{ifl,m} - (\Theta_{m,i+1} - \Theta_s) \cdot \Delta z / \Delta t \qquad \textbf{(2.14.22f)}$$

6.2 $\Theta_{m,i+1} < \Theta_r$ (reducing only outflows, this may happen especially if the time step is very long and there is a considerable amount of interflow or exfiltration into rivers)

a) q_m and q_{m+1} are flowing downwards (vertical throughflow), $q_{ifl} > 0$ or $q_{exf,m} > 0$
In a first step the outflow downwards is reduced:

$$q_{m,red} = q_m + (\Theta_r - \Theta_{m,i+1}) \cdot \Delta z / \Delta t \qquad \textbf{(2.14.23a)}$$

with $q_{m,red}$ reduced outflow from layer *m* to layer *m*+1 [m/s]
q_m unreduced outflow from layer *m* to layer *m*+1 [m/s]
Θ_r residual water content [-]
other variables like in equation (2.14.22a) to (2.14.22f)

If the sign of q_m is inverted by this reduction (if it will become positive: $q_m > 0$), the reduction itself is reduced. The remaining fraction of the reduction is then applied to the other possible outflows interflow and exfiltration (if they are positive; only one of these components can be there because interflow is generated from layers above the groundwater only and exfiltration is generated from layers within the groundwater only):

$$q_{exf,m,red} = q_{exf,m} - q_{m,red} \text{ and } q_{m,red} = 0 \text{ (if } q_{exf,m} > 0)$$
$$q_{ifl,m,red} = q_{ifl,m} - q_{m,red} \text{ and } q_{m,red} = 0 \text{ (if } q_{ifl,m} > 0) \qquad \textbf{(2.14.23b)}$$

with $q_{exf,m,red}$ reduced exfiltration from layer *m* [m/s]
$q_{exf,m}$ original exfiltration after equations (2.14.12) to (2.14.14) [m/s]
$q_{ifl,m,red}$ reduced interflow from layer *m* [m/s]
$q_{ifl,m}$ original interflow after equation (2.14.10) and (2.14.11) [m/s]

b) all Fluxes are outflows (q_m: ↓, q_{m-1}:↑, $q_{ifl} > 0$ or $q_{exf,m} > 0$)
in addition to q_m and $q_{exf,m}$ and q_{ifl} , q_{m-1} is reduced analog equation (2.14.23a) .

c) inflow from above and below but also outflow by q_{ifl} or $q_{exf,m}$ (q_m: ↑, q_{m-1}: ↓, $q_{ifl} > 0$)
only reduction of q_{ifl} or $q_{exf,m}$ after equation (2.14.23a) , replacing there q_m by q_{ifl} resp. q_{exf} .

d) all fluxes are directed upwards and there is interflow (q_m: ↑, q_{m-1}:↑, $q_{ifl} > 0$; this is rather improbable). In this case all outflows are reduced (q_{m-1}, q_{ifl} or $q_{exf,m}$) in analogy to equation (2.14.23a) /(2.14.23b) , the signs has to be changed according to the direction of flow.

7. *getting the iteration error and comparison with the break criterion*:

After calculating the fluxes the new water content values are calculated for all layers according to equation (2.14.7) . The maximum change of the water content of two successive iteration steps is compared with the allowed maximum error. If the change in water content is below this threshold for all layers, the scheme is assumed to be stable:

$$\epsilon_{max} = \max\left(\Theta_{m,iter} - \overline{\Theta_{m,iter-1}}\right) \qquad \textit{(for all m)} \qquad \textbf{(2.14.24)}$$

with ε_{max} maximum change of water content values for all layers between two iteration steps
$\Theta_{m,iter}$ water content after iteration step *iter* in the layer *m*
$\overline{\Theta_{m,iter-1}}$ accumulated average value of all water content values of the previous iteration steps within the actual time step. If *iter* < 4 then:
$\overline{\Theta_{m,iter-1}} = \Theta_{m,iter-1}$,
what means that the calculation is done not with the accumulated average but
with the real result from the last iteration step. In the first iteration step is $\overline{\Theta_{m,iter-1}} = \Theta_{m,i-1}$, the result from the last time step *i*-1

The iteration is broken if:

$$a)\ e_{max} < e_{zul}$$
$$b)\ iter \geq iter_{zul} \qquad \textbf{(2.14.25)}$$

with ε_{zul} maximum allowed error: $\varepsilon_{zul} = 10^{-5}$ for *iter* = 1…20, $\varepsilon_{zul} = 5 \cdot 10^{-5}$, for *iter* > 20
$iter_{zul}$ maximum allowed number of iteration steps (WaSiM internal: 100)

To avoid to much iteration cycles, the maximum allowed error is increased from $\varepsilon_{max} = 0.001\ \%_{Vol.}$ to $\varepsilon_{max} = 0.005\ \%_{Vol.}$ if the number of iteration steps *iter* is greater than 20. Usually this case occurs only for large time steps ($\Delta t > 1d$) when the FD-scheme tends to extreme states because of the strong non-linearities in the *k*-Θ- and ψ-Θ-relations. These non-linearities in conjunction with long time steps are also the reasons for using the accumulated averages for the soil water content values starting from the 4th iteration step instead of the iteration results itself to calculate the new *k* and ψ values.

2.14.4 Withdrawal of evaporating water

The extraction of water from the different soil layers is done separately for soil evaporation and for transpiration before calculating the soil water fluxes. The first extraction step is the extraction of transpiration water which in each layer may be decreased according to the soil moisture. The second step is the extraction of soil evaporation from bare soil. Also the evaporation may be reduced according to the moisture of the uppermost soil level. If the transpiration was reduced according to the soil moisture it is still possible to extract this remaining water (at least fractionally) from the uppermost layer by evaporation until the potential evaporation is satisfied.

1.calculating extraction by transpiration:

For each kind of vegetation there is a parameter for the root density distribution with depth. Another set of parameters describes the annual course of the root depth. The root density distribution can be described after GURTZ (1988) by:

$$d_r(z) = d_{norm}\left(1-\frac{z}{z_w}\right)\cdot\left(1+\frac{z}{z_w}\cdot\rho\right) \qquad \textbf{(2.14.26)}$$

with d_r root density at depth z
z depth below soil surface [m]
z_w maximum root depth (annual course specific for each crop type)
ρ parameter of the root density distribution ($\rho > -1$)
for $-1 < \rho < 0$ concave shape with extraction mainly from the upper layers
for $\rho = 0$ linear decrease of extraction with depth
for $\rho > 0$ convex shape, extraction more from the lower layers
d_{norm} scaling factor to scale the sums of the discrete extraction densities to 1.0

At first, the extraction density $d_{r,m}$ are calculated for the centers of all layers 1..m which are entirely within the root zone. If in the lowest root zone layer the roots does not reach the lower boundary of the layer, the center between the upper boundary $h_{geo,m-1}$ (expressed in meters below surface) and the maximum root depth z_w is taken for calculating the extraction density of this layer.

The second step is a check for possibly required reductions of evapotranspiration according to soil moisture. This is done starting at the top layer for each layer down to the lowest root zone layer. If an extraction from one layer is reduced, it is possible to compensate this reduction by increasing the extraction density of the next lower layer up to a certain limit:

$$\hat{d}_{r,m+1} = d_{r,m+1} + (1-x)\cdot d_{r,m} \qquad \textit{(for } (1-x)\cdot d_{r,m} \le \lambda\cdot d_{r,m+1}\textit{)} \quad \textbf{(2.14.27)}$$

with $\hat{d}_{r,m+1}$ increased extraction density for layer m+1 (increased by the reduction amount of the upper layer m)
$d_{r,m+1}$ original extraction density for layer m+1 after eq. (103)
$d_{r,m}$ extraction density for layer m (possibly also increased by reductions from upper layers)
λ maximum allowed factor for increasing the extraction density of layer m+1 in WaSiM set to 2.0 (this factor is ought to consider a balance between plant flexibility and limits of flexibility due to the given distribution of roots)
x reduction factor of the extraction density of layer m (0..1, 0 means no transpiration is taken from layer m, 1 means that no reduction at all is applied)

If the reduced amount of the extraction density (in terms of density!) in layer m is greater than λ times the extraction density of layer m+1, the extraction density of layer m+1 is increased only by

the maximum factor λ. The remaining amount of the reduction is then stored in a variable $E_{z,m}$ in terms of evaporation in mm and can be taken from the top soil layer by bare soil evaporation (if the total transpiration was less than the potential evaporation):

$$E_{z,m} = \left((1-x) \cdot d_{r,m} - (1+\lambda) \cdot d_{r,m+1} \right) \cdot ETP \cdot v \qquad \textit{(only for (1- x) } \cdot d_{r,m} > \lambda \cdot d_{r,m+1} \textit{)}$$

(2.14.28)

with $E_{z,m}$ additional available evaporation energy (because saved in layer *m*) for evaporation from the top soil [mm]
v vegetation coverage degree, crop specific annual course [-]
ETP potential evaporation [mm]

The extraction amounts for each soil layer are then calculated using the now corrected extraction densities, the vegetation coverage degree and the potential evaporation:

$$T_m = \hat{d}_{r,m} \cdot ETP \cdot v \qquad \textbf{(2.14.29)}$$

with T_m Transpiration from layer *m* [mm or m/s]
$\hat{d}_{r,m}$ corrected extraction density after equations (2.7.2) , (2.14.26) and (2.14.27)
ETP potential evaporation [mm or m/s]
v vegetation coverage degree [-]

2. evaporation from the top soil layer

Evaporation is only possible from the top soil and from the fraction of the area which is not covered by vegetation. If the default is used, the evaporation is taken from the uppermost soil layer only. However, this can be individually overridden by the new parameter EvapMaxDepth for each soil table entry. If the extractions from the soil were effectively reduced, the saved amount of energy E_z is still available to allow evaporation from the bare soil area. Like the transpiration, also the evaporation may be reduced according to the moisture Θ_1 within the top soil layer , but following another scheme:

$$E = \left[(1-v) \cdot ETP + v \cdot E_z \right] \cdot \left[1 - \left(\frac{\Theta_{s,1} - \Theta_1}{\Theta_{s,1}} \right)^2 \right] \qquad \textbf{(2.14.30)}$$

with E evaporation from the top soil layer [mm]
ETP potential evaporation (after reduction by the interception evaporation)
v vegetation coverage degree, crop specific annual course [-]
E_z sum of the uncompensated reductions after equation (2.14.27)
$\Theta_{s,1}$ saturation water content of the top soil layer (usually 1, hence index 1 here) [-]
Θ_1 actual water content of the top soil layer [-]

3. updating the soil water content

After extracting the evaporation and transpiration water from the soil layers 1..*m*, the water content is updated for all these layers:

$$\begin{aligned} \Theta_{m,i+1} &= \Theta_{m,i} + T_{m,i} \cdot d_m \cdot \Delta t^{-1} && \text{for } m > 1 \\ \Theta_{1,i+1} &= \Theta_{1,i} + (T_{1,i} + E_{1,i}) \cdot d_m \cdot \Delta t^{-1} && \text{for } m = 1 \end{aligned} \qquad \textbf{(2.14.31)}$$

2.14.5 Considering irrigation

Within WaSiM irrigation can be controlled by two different ways – by an irrigation schedule which contains the irrigation amount for each irrigation day as crop specific values or by a process

controlled by the model itself using soil moisture data and crop specific threshold values. For both methods the model needs information about the irrigated areas which is given by an irrigation grid. This grid contains for each irrigated crop type a specific code allowing a localization of irrigated areas. The control file contains an irrigation table which contains a corresponding irrigation description for each grid code from the irrigation grid.

- whether the irrigation is controlled by a schedule or by internal model states (soil moisture),
- whether the irrigation water is taken from the groundwater (from the same cell where irrigation is applied) or from surface water (from the same subbasin the irrigated cell is located in); the water balance is done in every case for groundwater and surface water, respectively
- if irrigation is controlled by a schedule: when is what crop type irrigated with how much water
- or, if irrigation is controlled by soil moisture: what is the threshold suction ψ_{dry} of each irrigated crop type for starting irrigation and what for stopping irrigation (ψ_{wet})

If the irrigation is controlled by soil moisture, the threshold suctions ψ_{dry} and ψ_{wet} are converted into corresponding water content values Θ_{dry} and Θ_{wet} which vary dependent on the soil type the crop is growing on. If the model enters the irrigation module, it compares the mean soil water content within the entire root zone with the lower threshold value Θ_{dry}. If this threshold is reached the irrigation amount I is calculated by:

$$I = (\Theta_{wet} - \Theta_{dry}) \cdot z_w \cdot 1000 \qquad \textbf{(2.14.32)}$$

with I irrigation amount [mm]
Θ_{wet} upper threshold for soil water content during irrigation [-]
Θ_{dry} lower soil water content for triggering the irrigation (lower threshold) [-]
z_w root depth [m]

The irrigation amount I is split into 24 fraction for the next 24 hourly time steps (if using a time step of one hour) to avoid to high irrigation intensities which could lead to surface runoff. The leakage fraction can not be specified exactly in the control file. To control the leakage fraction, the upper moisture threshold Θ_{wet} can be changed using the parameter ψ_{wet} in the control file. If ψ_{wet} is greater than the suction at field capacity (around 0.3 … 0.5 m) there will be no leakage but small values for ψ_{wet} of e.g. 0.1 to 0.2 m will lead to more ore less fast drainage (leakage).

The irrigation water is taken from groundwater or from surface water and is added to the precipitation. The tracer concentrations are mixed with the input concentrations from precipitation.

2.14.6 Considering artificial drainage

The effects of artificial drainage's is calculated separately and afterwards added to the interflow. The parameterization is done using two grids – a grid containing the depth z_d of the drainage tiles or hoses and another grid containing the horizontal spaces d_h between the drainage devices. If the drainage depth in the depth-grid is zero or the cell has nodata then no drainage is assumed there. The drainage water is added to the interflow of the same layer where the drainage system is located in:

$$q_{drain} = k(\Theta) \cdot \frac{2d_m}{d_h} \cdot \frac{cs}{d_h} \qquad \textbf{(2.14.33)}$$

with q_{drain} drainage from layer m, if drainage tiles or hoses are located in this layer [m/s]
$k(\Theta)$ hydraulic conductivity as a function of water content and soil type (2.14.5)
d_m layer thickness of the drainage layer m (discretization in z-direction) [m]
d_h horizontal spacing of the drainage tiles [m]
cs grid cell size [m]

The term $2d_m/d_h$ is the hydraulic gradient within the layer *m* which is directed to the drainage system. The term cs/d_h considers the drainage density in this cell thus making the algorithm scale independent. The drainage is added to the interflow of the same layer and thus included in the flux and gradient restrictions (especially the check after equations (2.14.10) and (2.14.11) must be done in order to get not too much drainage). Because the calculation of drainage is included in the iteration of the unsaturated zone, the drainage can affect also the downward fluxes from the layers above.

2.14.7 Considering clay horizons

Clay horizons or horizons with a significant lower conductivity may be generated by a combination of climate, humane impact (cropping technologies), or by the soil genesis. However, in WaSiM there is a possibility to consider a so called clay-grid which contains for each grid cell the depth of the upper limit of a clay layer. If this depth is zero or if the value is nodata no clay layer is assumed for this cell. If WaSiM can find this grid during initialization *and* if a flag in the control file is set to 1, then the hydraulic conductivity of that discretization layer which is containing the clay layer is replaced by a constant value given in the control file. This is only an extension of the parameterization of the soil in the control file and does not tangent the finite difference scheme for calculating the unsaturated flow.

2.14.8 Macropore infiltration

The macropore approach implemented into WaSiM follows the „bypass-flow-concept" after Jansson und Karlberg (2001). This approach allows the simulation of macropore runoff in layered soil profiles. The approach describes macropores by three parameters: depth of the macropores, capacity of the macropores and reduction of the macropore capacity per meter soil depth. By means of these parameters a maximum possible macropore flow can be calculated for each soil layer (potential macropore runoff). An additional parameter is given by the precipitation threshold value. Only if this predefined precipitation intensity is reached or exceeded, water can infiltrate into the macropores and macropore runoff can be generated. The real amount of macropore infiltration depends on the actual water content of the adjacent soil layers. Water which exceeds the free capacity of these soil layers, can not infiltrate into the macropores.

The infiltrated macropore water is used to fill up the adjacent soil layers from bottom to top. When an actual soil layer is saturated then the remaining macropore water will infiltrate into the next upper soil layer. Macropore water is not stored from one simulation time step to another but rather has to infiltrate completely into the soil layers within the actual time step. As the macropore infiltration is strongly dependent on the non-filled pore volume of the soil, the actual macropore runoff can be reduced with respect to the potential runoff considerably when the soil is (partly) saturated – thus it is common to have macropore runoff at the beginning of a heavy rainfall event rather.

The parametrization of the macropores is optional and done with the following parameters in the soil table. Missing entries are automatically replaced by the following values (i.e. deactivation of the macropore system):

- *PMacroThresh* = 1000 [mm/h]
- *MacroCapacity* = 0 [mm/h]
- *CapacityRedu* = 1.0 [1/m]
- *MacroDepth* = 1.0 [m]

2.14.9 Considering ponding water

Water ponding, e.g. at rice fields, can be considered by defining a pond depth grid which contains

for each grid cell the maximum storage capacity of the pond in m. If the capacity is zero or if the value is nodata, no ponding is assumed. If there is water in a pond, its amount is added to the precipitation, the tracer concentrations are mixed respectively added (in case of salt masses). Evaporation is taken from ponding water before starting the scheme of unsaturated zone flows, so there will be no transpiration and thus no change in the soil water content if there is enough water in the pond. This avoids numerical problems but another problem may persist: an enrichment of salt in the pond because there is no flow of water and salt into the soil (e.g. if the soil is unleaky because of a clay layer).

When using the lake model, the pond grid will become a special importance: lakes will be modelled by using the pond grid as the lake water level. Thus, the lake model can handle a lake like a flooded grid cell. For more information see the chapter on the lake model in this documentation.

2.14.10 Groundwater table elevation

The groundwater table is modeled in both the unsaturated zone module and the groundwater model. The coupling between both modules is done by a net boundary flux between the unsaturated zone and the groundwater. If the groundwater model calculates a change of the groundwater tables for a cell, the rate of change is converted into a vertical flux, positive for a falling groundwater table, negative for a rising groundwater table. This boundary flux and the amounts of infiltration, exfiltration, precipitation and fluxes between the unsaturated layers are considered to estimate the groundwater table. Assuming that the modeled soil column reaches down to the groundwater the groundwater table is located in the lowest partly saturated layer. Its "exact" location is interpolated. Using equation (2.14.6) it is possible to calculate the water content a layer must have if being in a hydrostatic equilibrium with the groundwater at the lower layer boundary.

$$\Theta_{GW,\min} = \Theta_r + (\Theta_s - \Theta_r)\left(\frac{1}{1+(0.5 \cdot d_m \alpha)^n}\right)^m \qquad \textbf{(2.14.34)}$$

with $\Theta_{GW,min}$ water content for hydrostatic equilibrium with the groundwater table (referring to the center of the layer)
Θ_s saturation water content of layer m [-]
Θ_r residual water content of layer m [-]
α, m, n Van Genuchten - parameter
d_m layer thickness of layer m [m]

The actual groundwater table is then interpolated after each time step:

$$h_{GW} = h_{\text{geo},m} ¿ + d_m\left(\Theta_{m,i} - \Theta_{GW,\min}\right)\left(\Theta_s - \Theta_{GW,\min}\right) \qquad \textbf{(2.14.35)}$$

with h_{GW} groundwater level [m a.s.l.]
$h_{geo,m}$ geodetic altitude of the lower limit of the lowest partly saturated layer [m]
d_m thickness of layer m [m]
$\Theta_{m,i}$ actual water content of layer m at time step i
Θ_s saturation water content [-]

This equation (2.14.35) is valid only if the groundwater table is moving within a single layer. If the groundwater table moves to another layer there must de differentiated between two possibilities:

1.dropping down to the next lower layer: The equilibrium water content $\Theta_{GW,min}$ must be newly calculated for this next layer

2.entering the upper layer: As long as the groundwater table rises, the last water content Θ_{m-1} of the layer m-1 is stored as $\hat{\Theta}_{GW,\min}$ and used for the interpolation of the groundwater table. Only if the groundwater table is falling again *and* at the same time $\hat{\Theta}_{GW,\min}$ is greater than $\Theta_{GW,\min}$

(calculated after equation (2.14.34)), the value o $\hat{\Theta}_{GW,\min}$ f has to be adopted to $\Theta_{GW,\min}$. This is done by adding the balance of inflows and outflows to the actual value of $\hat{\Theta}_{GW,\min}$:

$$\Delta\Theta_{m,i} = q_{m-1} + Q_B + Q_{ifl} + ETR_m$$
$$\hat{\Theta}_{GW,min,i+1} = \hat{\Theta}_{GW,min,i} + \Delta\Theta_{m,i} \quad \text{for } \Delta\Theta_{m,i} > 0 \text{ (falling groundwater table)} \qquad \textbf{(2.14.36)}$$

with $\Delta\Theta_{m,i}$ change of water content of layer *m* during time step *i* [-]
q_{m-1} inflow from layer *m*-1 (may be positive or negative!) [m/s]
Q_B lateral net runoff of layer *m* as given by the groundwater model [m/s]
Q_{ifl} interflow from layer *m* [m/s]
ETR_m extraction by transpiration from layer *m* [m/s]

If the value of $\hat{\Theta}_{GW,\min}$ becomes smaller than the value of $\Theta_{GW,min}$, the latter value is taken for the interpolation of the groundwater table and $\hat{\Theta}_{GW,\min}$ is not considered anymore until the groundwater is entering the layer again from below.

2.14.11 Groundwater recharge

The groundwater recharge is the balance of inflows and outflows to that layer which contains the groundwater level. Recharge can be positive or negative. If the groundwater tables moves to another layer, the balances are calculated including the fluxes of all affected layers. The drained or filled porosity is balanced with the lateral net flux change from the groundwater model (the rate of rise or fall of groundwater in terms of mm/time step). There are 4 cases for calculating the drained or filled porosity:

1.groundwater table is falling within the same layer:

The layer is split into an unsaturated and a saturated fraction. The thickness d_u of the unsaturated fraction is given by $d_u = h_{geo,m-1} - h_{GW}$ (see equation (2.14.30)):

$$\Theta_{u,new} = \frac{\left(d_m\Theta_{m,i} - (h_{GW,i} - h_{geo,m})\Theta_s\right)}{h_{geo,m-1} - h_{GW}} \qquad \textbf{(2.14.37)}$$

with $\Theta_{u,new}$ water content of the unsaturated fraction of layer *m* (the layer the groundwater table is within)
d_m thickness of layer *m* [m]
$h_{GW,i}$ groundwater table at time step *i*
$\Theta_{m,i}$ average water content of layer *m* in the actual time step *i*

The water content for the lower fraction between h_{GW} and $h_{geo,m}$ is at saturation: $\Theta = \Theta_s$. The drainable porosity is thus given by:

$$\Theta_e = (h_{GW,i} - h_{GW,i-1}) \cdot (\Theta_s - \Theta_{u,new}) \qquad \textbf{(2.14.38)}$$

with $h_{GW,i}$ groundwater table in the actual time step *i* [m]
$h_{GW,i-1}$ groundwater table from the previous time step *i*-1 [m]

2. groundwater table is rising within the same layer

$$\Theta_{u,old} = \frac{\left(d_m\Theta_{m,i-1} - (h_{GW,i-1} - h_{geo,m})\Theta_s\right)}{h_{geo,m-1} - h_{GW,i-1}} \qquad \textbf{(2.14.39)}$$

with $\Theta_{u,old}$ water content of the unsaturated fraction of layer *m* from the previous time step *i*-1 [-]
d_m thickness of layer *m* [m]

$h_{GW,i-1}$ groundwater table from the previous time step *i*-1 [m]
$\Theta_{m,i-1}$ average water content of layer *m* from the previous time step *i*-1 [-]

The filled porosity is then given by:

$$\Theta_e = (h_{GW,i} - h_{GW,i-1}) \cdot (\Theta_s - \Theta_{u,old}) \quad \textbf{(2.14.40)}$$

with $h_{GW,i}$ groundwater table in the actual time step *i* [m]
$h_{GW,i-1}$ groundwater table from the previous time step *i*-1 [m]

If the groundwater table moves into another layer (over one or more layer boundaries), the results of equations (2.14.38) and (2.14.40) have to be extended by the drained or filled porosities of the completely drained or filled layers:

3.groundwater is moving down a layer limit :

$$\Theta_e = \left(h_{geo,n} - h_{GW,i-1}\right) \cdot \left(\Theta_s - \Theta_{n,i}\right) + \sum_{k=n+1}^{m} \left(d_k \left(\Theta_{k,i-1} - \Theta_{k,i}\right)\right) \quad \textbf{(2.14.41)}$$

with $h_{geo,n}$ geodetic altitude of the layer the groundwater table was within in time step *i*-1
$h_{GW,i-1}$ groundwater table in the previous time step *i*-1
$\Theta_{n,i}$ actual water content in the layer *n* the groundwater table is within in time step *i*
n *i*ndex: layer the groundwater was within in time step *i*-1
m index: layer, the groundwater is within in time step *i*
d thickness of the layers [m]
k loop index for all completely drained layers

4.groundwater table rises above a layer limit:

$$\Theta_e = \left(h_{GW,i} - h_{geo,m}\right) \cdot \left(\Theta_s - \Theta_{m,i} - \Delta t \cdot Q_{lat} / d_m\right) + \sum_{k=n+1}^{m} \left(d_k \left(\Theta_{k,i} - \Theta_{k,i-1}\right)\right) \quad \textbf{(2.14.42)}$$

with $h_{geo,m}$ geodetic altitude of the layer the groundwater table is within in time step *i*
$h_{GW,i}$ groundwater table in the actual time step *i*
$\Theta_{m,i}$ actual water content in the layer *m* the groundwater is within in time step *i*
n index: layer the groundwater was within in time step *i*-1
m index: layer, the groundwater is within in time step *i*
d thickness of the layers [m]
Δt time step [s]
Q_{lat} lateral Net flow in time step *i*
k loop index for all completely filled layers

The groundwater recharge is for all 4 cases the sum of the drained respectively the filled porosity Θ_e and the lateral net fluxes of the previous time step:

$$GWN = 1000 \cdot \Theta_e + Q_{B,i-1} \quad \textbf{(2.14.43)}$$

with *GWN* groundwater recharge [mm]
Θ_e filled porosity (positive sign) resp. drained porosity (negative sign) [-]
$Q_{B,i-1}$ base flow/lateral net flow from the previous time *i*-1 [mm]

2.14.12 Calculating baseflow

Base flow is generated in WaSiM version 2 only as exfiltration from the groundwater into the surface river system. It is generated only at that grid cells which are marked in a number of grids as river cells. The output is written not only as a grid but also as an areal average value for subbasins. For this output files the exfiltration into rivers is not balanced with the infiltration from rivers into

the groundwater. This is done during the discharge routing.

However, if the model is run without the groundwater module, calculation of base flow must be done in a conceptual way, because no lateral exchange between cells is possible. For such cases an approach similar to the TOPMODEL-Approach is used:

$$Q_B = Q_0 \cdot K_s \cdot e^{(h_{GW} - h_{geo,0})/k_B} \quad \textbf{(2.14.44)}$$

with Q_B base flow [m/s]
Q_0 scaling factor for base flow (or maximum baseflow if the soil is saturated) [-]
K_s saturated hydraulic conductivity [m/s]
h_{GW} groundwater table [m a.s.l.]
$h_{geo,0}$ geodetic altitude of the soil surface (upper limit 1st layer) [m a.s.l.]
k_B recession constant for base flow [m]

This approach is applied to each cell of the model grid – base flow is thus not generated only at river cells. The parameters Q_0 and k_B have to be calibrated.

2.14.13 Coupling with the groundwater model

Starting from the interpolation all model components including snow, evaporation, interception, irrigation and unsaturated zone were only one dimensional. The generation of interflow is an extension which presumes a sufficiently larger cell size than the natural drainage systems density (small drainage channels can be found usually after at least 10 to 100 meters in hilly regions). Neighboring cells does not communicate in this scheme, otherwise a complete 3-D-model for the unsaturated zone would be required. On the other hand, coupling a horizontally 2-dimensional groundwater model to the unsaturated zone is very easy if the groundwater flux field is calculated independent on the unsaturated zone.

The coupling in WaSiM (version 2) is done bi-directional.

- The unsaturated zone module calculates the flux between the unsaturated zone and the groundwater. This flux is given to the groundwater module as upper boundary condition (known flux boundary). Tracer concentrations of the net fluxes are balanced with the groundwater and the unsaturated zone.

- The groundwater model calculates the lateral fluxes using the boundary fluxes from the unsaturated zone model. The net change of the groundwater table is converted into a boundary flux (in mm/time step vertical flux) and given to the unsaturated zone to be considered as inflow or outflow.

2.14.14 Dynamic time step control

The implementation of a dynamic time step controller into the Richards-version of WaSiM allows a more flexible and process-oriented soil parametrization as well as an improved simulation quality of vertical water fluxes in layered soils compared to a uniform internal time step. Using this controller, soil water fluxes can be simulated with a very high level of numerical stability even in cases with extreme parameter settings (e.g. very thin soil layers combined with the pronounced hydraulic conductivity of sandy soils).

The introduced dynamic time step control algorithm (available in WaSiM since version 7.9) considers explicitly the Courant stability criterion in order to prevent oscillations possibly caused by the discretization scheme of the actual soil profile. The Courant criterion is defined as follows:

$$Co = \left| \frac{\Delta t \cdot v}{\Delta x} \right| \leq 1 \quad \textbf{(2.14.45)}$$

with Co Courant criterion
Δt time step [s]
v velocity [m/s]
Δx spatial distance between the nodes [m]

After its transformation the boundary condition for a stable numerical solution are given by:

$$\Delta x \geq \Delta t \cdot v \qquad \text{resp.} \qquad \Delta t \leq \frac{\Delta x}{v} \tag{2.14.46}$$

with Δt simulation time step [s]
v hydraulic conductivity [m/s]
Δx thickness of the actual soil layer [m]

In WaSiM the algorithm of dynamic time step control has been a slightly adapted:

$$\Delta t = MIN_{i=1}^{k}\left(\frac{\Delta z_i}{k(\Theta)_{rel,i} \cdot k_{sat,i}}\right) \tag{2.14.47}$$

with Δt minimum time for flowing of water through the soil layer at the actual water content [s]
i layer index (from 1..k)
k index of last soil layer above the groundwater level
Δz_i thickness of the actual soil layer i [m]
$k(\Theta)_{rel,i}$ relative moisture-dependent conductivity of layer i
$k_{sat,i}$ saturated hydraulic conductivity of layer i [m/s]

If the Courant time interval is shorter than the predefined model time step ($\Delta t < I$) then a certain split factor has to be calculated in order to reduce the length of the original time step:

$$f_{split} = \left\lceil \frac{I}{\Delta t} \right\rceil \tag{2.14.48}$$

with f_{split} split factor with $f_{split} >= 1$
I length of the selected time step interval (as defined in control file) [s]
Δt result from Equation (2.14.11) [s]
$\lceil\rceil$ character for a rounded up integer

The internal reduction of the (original) model time step is done separately for each grid cell. For this purpose the implemented control algorithm checks all soil layers above the groundwater level with regard to the actual Courant criterion. If necessary ($\Delta t < I$) then the (original) model time step is subdivided into a certain number of subintervals (f_{split}). In this case, the absolute input data [in mm] for the soil model have to be recalculated for each subinterval whereas data of flow velocity (e.g. descent and ascent rates of the groundwater level or water flows between soil layers) can be used as before (i.e. without subdivision) because of their independence of the length of calculation time interval.

The shortest possible subinterval is one second. That means in effect that a model time step of one hour can be subdivided into up to 3600 subintervals. In order to increase the flexibility of the implemented algorithm, it is possible to define a minimum subinterval length [in s].

The dynamic time step control can be activated by selecting method “3” in the section

[unsatzon_model] of the WaSiM control file.

2.14.15 Control file extensions for the unsaturated zone model

The following control file sequence was taken from a “real world“ example – the River Rhone basin in Switzerland. It is only slightly simplified for better clarity (rows ending with ↵ have a line break for this documentation only. In real control files, the single parameters must follow each other without empty lines and without comments or lines starting without #):

```
[unsatzon_model]
1     # 0=ignore this module, 1 = run the module
$time # duration of a time step in minutes
3     # method, 2 = FDM-Method (fix step) 3 = FDM-Method with dynamic time step
1     # controlling interaction with surface water: 0 = no interaction, ↵
      1 = exfiltration possible 2 = infiltration and exfiltration possible
1     # controlling surface storage in ponds: 0 = no ponds, 1 = using ponds ↵
      for surface storage (pond depth as standard grid needed -> height of ↵
      dams oround fields)
0     # controlling artificial drainage: 0 = no artificial drainage ↵
      1 = using drainage (drainage depth and horizontal pipe distances as ↵
      standard grids needed!)
0     # controlling clay layer: 0 = no clay layer,  1 = assuming a clay layer ↵
      in a depth, specified within a clay-grid (declared as a standard grid)
5e-8  # permeability of the clay layer (is used for the clay layer only)
4     # parameter for the initialization of the gw_level ↵
      range between 1..levels (standard: 4). 4 means: the upper 1/4th ↵
      of the layers is unsaturated
qdra//$grid//.//$year $sum_per_24Invs  # drainage in mm per zone
gwst//$grid//.//$year $avrg_per_24Invs # groundwater depth
gwn_//$grid//.//$year $sum_per_24Invs  # groundwater recharge per zone
sb05//$grid//.//$year $avrg_per_24Invs # rel. soil moisture in root zone
sb1_//$grid//.//$year $avrg_per_24Invs # rel. soil moisture within the unsat.↵
                                        zone (0m..GW table) per zone
wurz//$grid//.//$year $avrg_per_24Invs # root depth
infx//$grid//.//$year $sum_per_24Invs  # infiltration excess
pond//$grid//.//$year $avrg_per_24Invs # ponding water storage content
qdir//$grid//.//$year $once_per_interval  # direct discharge
qifl//$grid//.//$year $once_per_interval  # interflow
qbas//$grid//.//$year $once_per_interval  # baseflow
qges//$grid//.//$year $once_per_interval  #total discharge
gwin//$grid//.//$year $once_per_interval  # infiltration from surface water ↵
                                    into groundwater (from rivers and lakes)
gwex//$grid//.//$year $once_per_interval  # exfiltration from groundwater ↵
                                    into surface water (into rivers and lakes)
macr//$grid//.//$year $sum_per_24Invs     # infiltration into macropores
qinf//$grid//.//$year $once_per_interval  # total infiltration into the first ↵
                                        soil layer
$SB_1_grid  # grid with actual soil water content for the root zone
$Writegrid  # Writecode for this grid
$SB_2_grid  # grid with actual soil water content entire unsaturated zone
$Writegrid  # Writecode for this grid
$ROOTgrid   # grid with root depth
$Writegrid  # Writecode for this grid
$Thetastack # stack, actual soil water content for all soil levels
$Writegrid  # Writecode for this stack
$hydraulic_heads_stack # stack, contaiing hydraulic heads
$Writestack            # Writecode for this stack
$geodetic_altitude_stack # stack, geodaetic altitudes of the soil levels
$Writestack              # Writecode for this stack
$flowstack       # stack, containing the outflows from the soil levels
```

```
$Writestack       # Writecode for this stack
$GWdepthgrid      # grid with groudwaterdepth
$Writegrid        # Writecode for this grid
$GWthetagrid      # grid with theta in GWLEVEL
$Writegrid        # Writecode for this grid
$GWNgrid          # grid with groundwater recharge
$Writegrid        # Writecode for this grid
$GWLEVELgrid      # grid with level index of groundwater surface
$Writegrid        # Writecode for this grid
$QDRAINgrid       # grid with the drainage flows
$Writegrid        # Writecode for this grid
$SATTgrid         # grid with code 1=saturation at interval start, 0 no sat.
$Writegrid        # Writecode for this grid
$INFEXgrid        # grid with infiltration excess in mm (surface discharge)
$Writegrid        # Writecode for this grid
$QDgrid           # grid with direct discharge
1//$Writegrid     # Writecode for this grid
$QIgrid           # grid with Interflow
1//$Writegrid     # Writecode for this grid
$QBgrid           # grid with baseflow
1//$Writegrid     # Writecode for this grid
$GWINgrid         # grid with infiltration from rivers into the soil/groundwater
1//$Writegrid     # Writecode for this grid
$GWEXgrid         # grid with exfiltration (baseflow) from groundwater ↲
                   (is only generated, if groundwater module is active, ↲
                  else baseflow is in QBgrid)
1//$Writegrid     # Writecode for this grid
$act_pond_grid    # grid with content of ponding storge
$Writegrid        # Writecode for this grid
$UPRISEgrid       # grid with amount of capillary uprise (mm)
1//$Writegrid     # Writecode for this grid
$PERCOLgrid       # grid with amount of percolation (mm)
1//$Writegrid     # writegrid for this grid
$MACROINFgrid     # grid with amount of infiltration into macropores (mm)
1//$Writegrid     # Writecode for this grid
$irrig_grid       # grid with irrigation amount (will be written when ↲
                    irrigation is used, only)
$Writegrid        # writegrid for this grid (will be written when ↲
                    irrigation is used, only)
70 50              # coordinates of control plot, all theta and qu-values ↲
                   are written to files (qu.dat, theta.dat in the directory, ↲
                   from which the model is started)
qbot//$grid//.//$year          # name of a file containing the vertical flows ↲
                          between the layers of the control point
thet//$grid//.//$year          # name of a file containing the soil moisture ↲
                          as theta values of the layers of the control point
hhyd//$grid//.//$year          # name of a file containing the hydraulic head ↲
                          of the layers of the control point
otherdata//$grid//.//$year           # name of a file containing some other ↲
                               water balance data of the control point ↲
                               (non layer data)↲
etrd//$grid//.//$year          # name of a file containing the withdrawal of soil
↲
                          water for each layer for the control point ↲
                          (due to transpiration)
intd//$grid//.//$year          # name of a file containing the interflow for the
↲
                          soil layers of the control point
1 2 3 4 5 6 7 8 9 10     # range for subbasin codes
$kd1  $kd2  $kd3  $kd4  $kd5  $kd6  $kd7  $kd8  $kd9  $kd10 # kd
$ki1  $ki2  $ki3  $ki4  $ki5  $ki6  $ki7  $ki8  $ki9  $ki10 # ki
$dr1  $dr2  $dr3  $dr4  $dr5  $dr6  $dr7  $dr8  $dr9  $dr10 # dr
```

```
0.4   0.4   0.4   0.4   0.4   0.4   0.4   0.4   0.4   0.4   # k in qb = ↲
                          Q0 * exp(-k/z) with z = depth to groundwater
0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   # Q0 in the ↲
                                                              above formula
$sdf1 $sdf2 $sdf3 $sdf4 $sdf5 $sdf6 $sdf7 $sdf8 $sdf9 $sdf10 # fraction ↲
                                                 of surface runoff on snow melt
$readgrids  # meanings are extended now! read the follwing comments ↲
storage_richards.ftz! if readgrids = 1, then this file contains the contents ↲
of the flow travel time zones for interflow and surface flow and for ↲
the tracers. If code is 2, then only the groundwater table grid (gwst*) is ↲
read in and all other grids will internally initialized from this one grid
300 # minimum dynamic time step in seconds. the smaller this number, the ↲
      longer the model runs but the results will be more accurate due to a ↲
      better maintained Courant condition
step//$grid//.//$year $avrg_per_24Invs    # statistic of the number of substeps
$SUBSTEPSgrid # grid with number of substeps
$Writegrid    # writecode for substeps, the areal distribution is of interest
for the annual average value. This is code 6 as first digit in 2-digit codes.
Or 5 for the entire model period
```

2.14.16 Optional Grids and Statistics for Field Capacity

For low flow calculations and droughts, a set of some optional grids and statistics for the usable soil water content could be activated – the so called uFC grids (usable field capacity, even if the value is not field capacity but often only a fraction of it).

Field capacity is the water holding capacity of a soil that was heavily irrigated or exposed to saturating rain when most of the excess water has drained away and the rate of drainage has decreased drastically (e.g. after 1 or 2 days). It's usually not a constant for a given soil, but in WaSiM the soil doesn't change it's properties, so for constant root depth over time, the field capacity value is constant as well. In WaSiM, the newly added (optional) field capacity grids are root depth depending and do not consider the residual water content, and are thus called the usable Field Capacity grids (uFC).

Physically, a certain suction should be defined in order to be able to calculate the water content at field capacity. Often, the field capacity is assumed to be the water content at a suction of ~ -33 kPa (=330hPa or 330 cm suction). However, WaSiM uses a default value of -20 kPa or 200hPa = 200cm suction. This value can be specified for each soil layer in the soil-table as shown in the following example:

```
[soil_table]
40
#co- name of the  # van Genuchten Parameter nach Carsel & Parrish (1988)
#de  soil profile
#-- ---------------
2     Silty_Clay_shallow {
                  method            = MultipleHorizons;
                  ThicknessScaling  = $ThicknessScaling1 ;
                  MaxInterflowDepth = $MaxInterflowdepth1 ;
                  PMacroThresh      = $PMacroThresh1;
                  MacroCapacity     = $MacroCapacity1;
                  CapacityRedu      = $CapacityRedu1;
                  MacroDepth        = $MacroDepth1;
                  horizon           = 1     2         ;
                  Name              = T_Cu  Sub_Cu    ;
                  ksat              = $2e7  $1.4e7    ;
                  k_recession       = 0.95  0.95      ;
                  theta_sat         = 0.43  0.40      ;
                  theta_res         = 0.11  0.12      ;
```

```
            (hPa) psi_FC            = 63    63        ;
                  alpha             = 1.7   1.7       ;
                  Par_n             = 1.5   1.5       ;
                  Par_tau           = 0.5   0.5       ;
                  thickness         = 0.10  0.15      ; #2m
                  layers            = 10    10        ;
            }
22    Silty_Clay_medium {
                  method            = MultipleHorizons;
                  ThicknessScaling  = $ThicknessScaling2 ;
                  MaxInterflowDepth = $MaxInterflowdepth2 ;
                  PMacroThresh      = $PMacroThresh2;
                  MacroCapacity     = $MacroCapacity2;
                  CapacityRedu      = $CapacityRedu2;
                  MacroDepth        = $MacroDepth2;
                  horizon           = 1     2               ;
                  Name              = T_Cu  Sub_Cu    ;
                  ksat              = $2e7  $1.4e7    ;
                  k_recession       = 0.95  0.95      ;
                  theta_sat         = 0.43  0.40      ;
                  theta_res         = 0.11  0.12      ;
            (hPa) psi_FC            = 63    96        ;
                  alpha             = 1.7   1.7       ;
                  Par_n             = 1.5   1.5       ;
                  Par_tau           = 0.5   0.5       ;
                  thickness         = 0.2   0.3       ; #4m
                  layers            = 10    10        ;
            }
...
```

If the psi_FC parameter is omitted, 200 is taken as default value.

To activate the uFC grids, four additional standard grids have to be defined in the [standard_grids] section as shown below:

```
# define variables for Field caapcity grid-names
$set $nFK_grid_mm_min      =  uFCminmm//$grid//.//$suffix # min usable water ↵
   content (i.e. residual water content) in mm
$set $nFK_grid_mm_max      =  uFCmaxmm//$grid//.//$suffix # max field ↵
   capacity in mm
$set $nFK_grid_mm          =  uFCmm//$grid//.//$suffix   # usable water ↵
   content in mm
$set $nFK_grid_perc        =  uFCperc//$grid//.//$suffix # usable water ↵
   content as perc of max


[standard_grids]
29 # number of standard grids
$inpath_grid//$nFK_grid_mm_min minFieldCapacity_mm  fillcode = 1 defaultValue ↵
    = 200 readcode = 0 outname = $outpath//$nFK_grid_mm_min   writecode = 103
# minimum usable field capacity in mm (depends on soil property psi_FC, or
# 200cm suction as default, calculated over the actual root depth at every time
# step, result of min grid is available when writecode contains a min output
# like 1xx)
$inpath_grid//$nFK_grid_mm_max maxFieldCapacity_mm  fillcode = 1 defaultValue ↵
    = 200 readcode = 0 outname = $outpath//$nFK_grid_mm_max   writecode = 203
# maximum usable field capacity in mm (depends on soil property psi_FC, or
# 200cm suction as default, calculated over the actual root depth at every time
# step, result of max grid is available when writecode contains a max output
# like 2xx)
$inpath_grid//$nFK_grid_mm usableFieldCapacity_mm  fillcode = 1 defaultValue ↵
```

```
    = 100 readcode = 0 outname = $outpath//$nFK_grid_mm   writecode = 5 ↵
    statfile = $outpath//uFCmm//$grid//.//$code//$year statcode = $hour_mean
# actual usable water capacity in mm (depends on max field capacity and the
# actual water content), including statistics output
$inpath_grid//$nFK_grid_perc usableFieldCapacity_perc  fillcode = 1 ↵
    defaultValue = 100 readcode = 0 outname = $outpath//$nFK_grid_perc ↵
    writecode = 5 statfile = $outpath//uFCperc//$grid//.//$code//$year ↵
    statcode = $hour_mean
# actual usable field capacity in % (depends on max field capacity and the
# actual water content), including statistics output
```

Note, that the parameters are written in one single line for each grid and that the ↵ sign stands for "line continues...".

minFieldCapacity_mm:

This is the residual water content in mm over the total root depth. The minimum is NOT calculated by the soil module but by statistics routines that are called for each standard grid by default if the grid write code is set to 1xx, (e.g. 103). In that case, a minimum grid is created at the end of the model period, containing for each cell the minimum value that was calculated during the model run, regardless in which time step this happened.

maxFieldCapacity_mm:

This is the maximum of the usable field capacity for each grid cell during the model run in mm. Again, the maximum is not calculated in the soil model but in the statistics routines that are called by default for each grid if a write code of 2xx is set (e.g. 203). The usable field capacity defines the total water content between the residual water content and the water content at field capacity in mm over the root zone and is a theoretical value (may in some model runs never be reached). When root depth is not constant, the value may change over time according to the soil properties of the layers with roots in it and their thicknesses (or a fraction of the lowest layer).

usableFieldCapacity_mm:

This is the actual difference between the actual water content and the residual water content for the entire root zone. If the actual water content is higher than at the theoretical field capacity (e.g. fully saturated), then the value may even be higher than the maxFieldCapacity_mm value. Values below 0 will be set to 0 (e.g. when the soil dries out due to evaporation below the residual water content).

usableFieldCapacity_perc:

This is the same data as the usableFieldCapacity_mm grid, mut expressed as percentage of the maxFieldCapacity_mm. Values of > 100% are cut and set to 100% (can happen when the water content is above field capacity).

Some sample grids are shown in figures 2.14.16.1 to 2.14.16.4. The shown basin is the Swiss part of the river Inn as well as three smaller basins that do not belong to the Inn but to other rivers but are still part of Switzerland. Figure 2.14.16.1 shows the residual water content for the root zone in mm. It is of course highest in the valley bottoms, where there is a thicker soil than in the high mountain regions (where there is often no soil at all or it may even be glacierized). Also, the root depth if higher (i.e. deeper) for pine forest, growing in the lower regions, than for grass or even rock (which has a virtual root depth as well, but it's usually only a few cm).

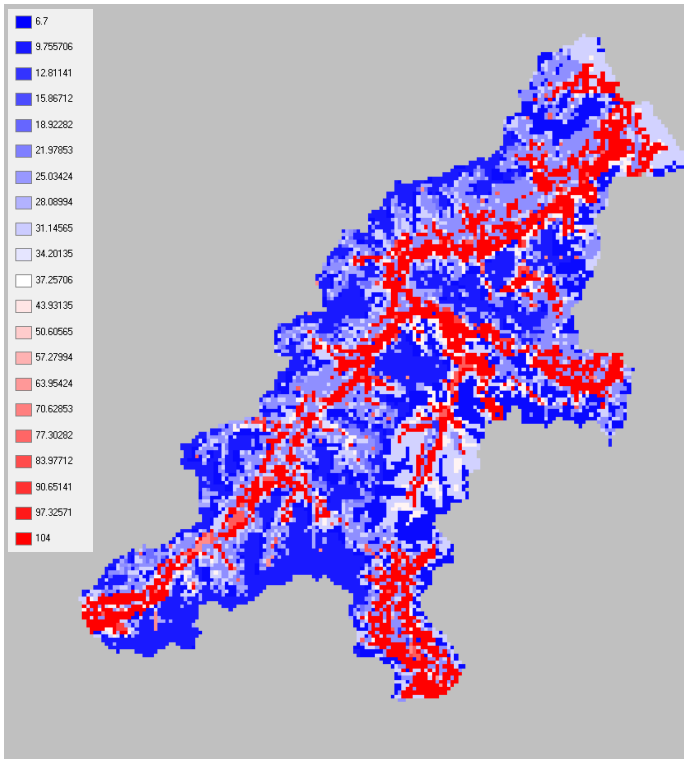


figure 2.14.16.1: Residual water content in minFieldCapacity_mm

Figure 2.14.16.2 shows the maximum field capacity values for the same basin. The picture looks quite different. For maximum field capacity, not only the root depth is important but also the soil type. The highest values can be seen where the root depth is high and the soil has a high absolute pore volume and at the same time the drainage is not good, i.e. a lot ow water can be hold back by the soil (silty loams for instance).

Figure 2.14.16.3 shows the same basin for January 31st, 1991. The pattern is similar to that shown in 2.14.16.2 but with another scaling. In particular, there are some spots that actually hold more water than the maximum field capacity defines (which is perfectly possible when the soil is saturated or has a suction that is smaller than the defined psi_FC parameter).

Corresponding to that actual field capacity in mm, figure 2.14.16.4 shows the same basin for the same day but in percentage of the maximum field capacity. As mentioned above, the percentage is cut at 0% and 100%, so there are a lot of cells with 100% which are higher saturated than the maximum field capacity would suggest (which is, again, perfectly possible).

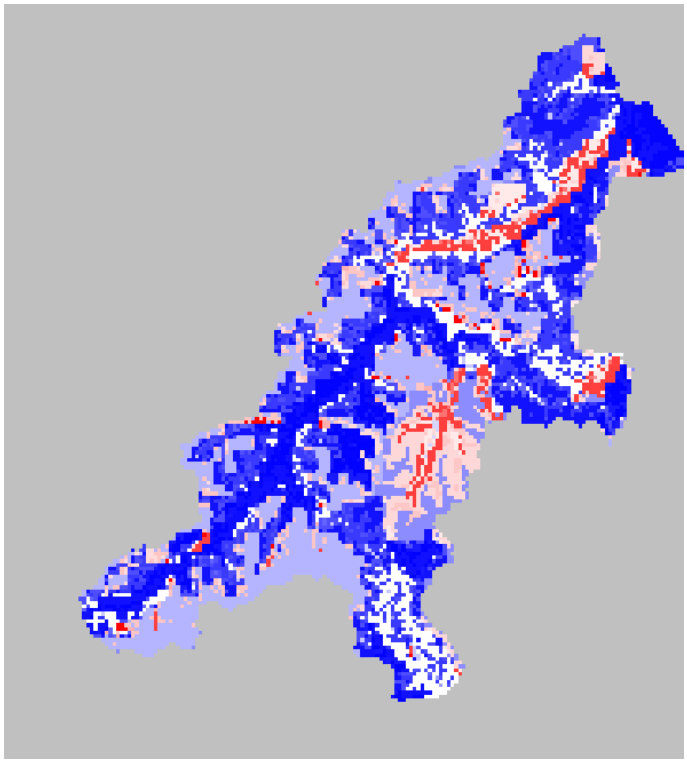


figure 2.14.16.2: Maximum usable field capacity in mm

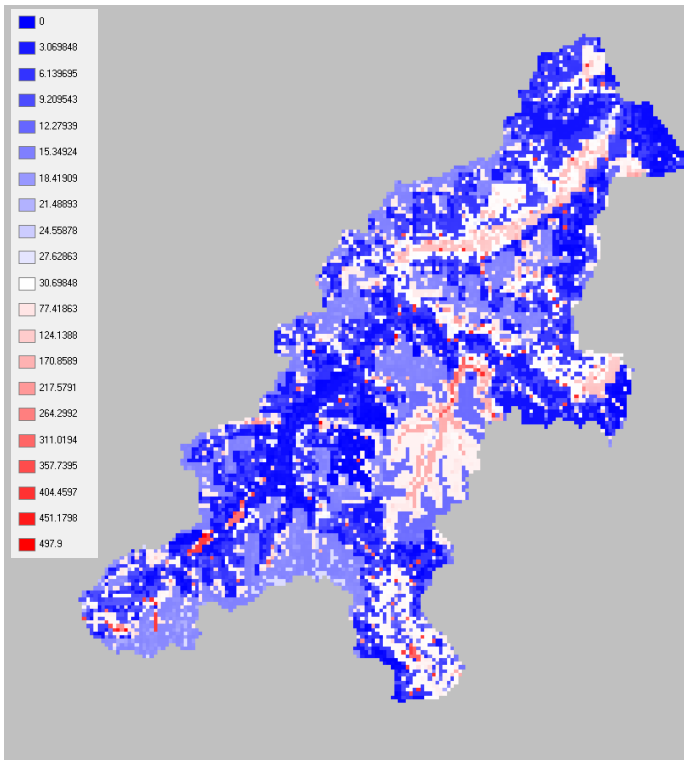


figure 2.14.16.3: actual field capacity 1991-01-31 in mm

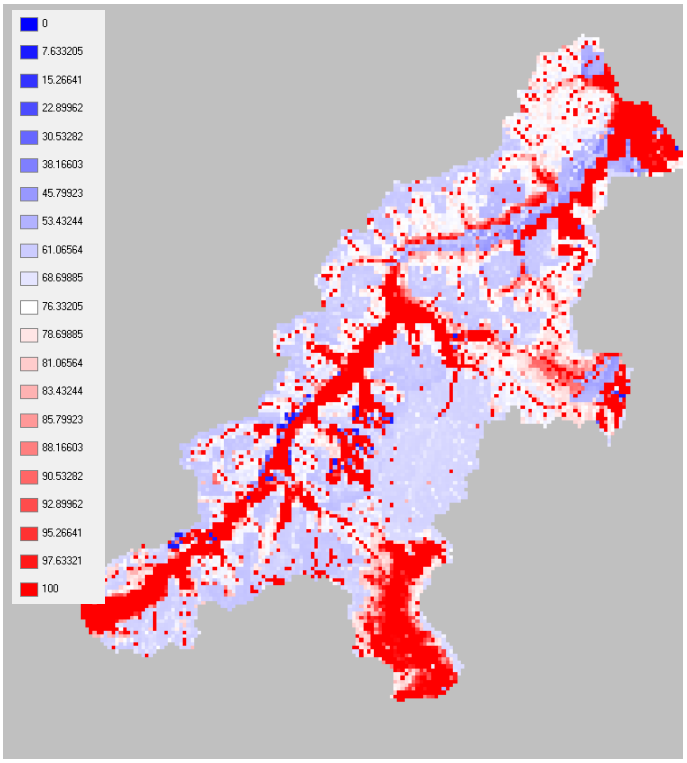


figure 2.14.16.4: actual water content 1991-01-31 as percentage of actual content on field capacity (in %)

2.14.17 A global parameter for defining the usable field capacity

In 2024, an extension of WaSiM was already made, in which it is possible to define a suction stress in cm (=hPa) for each individual horizon for each soil type, which corresponds to the water content at field capacity for this horizon. This makes it possible to define otherwise identical soils with different field capacities, depending on whether these soils are far from groundwater or close to groundwater.

However, since it can be tiresome to change all the values manually (although search/replace could help here as long as all values are identical), the possibility of an easy overruling was implemented in additiion to the existing horizon specific parameterization.

That overruling by a new global suction stress parameter can be used to quickly check changes in the time series of usable field capacity when the value of the threshold suction changes. Of course, the model needs to be run for the desired model period again after each change of that new parameter.

The configuration is quite easy:

There was a new optional section [GlobalSoilParameters] introduced. This section is intended to be extensible for future additional parameters. Therefor all parameters are written in the format

<key>=<value>;

```
GlobalSoilParameters]
psi_FC_global = 110; (hPa) # if this value is set, all soil types will use this value
for all horizons, except when usePsiFCglobalForUnassignedOnly = 1;
usePsiFCglobalForUnassignedOnly = 0; # if 0: psi_FC_global is used for all soil types
and horizons, whether the local value of psi_FC is set or not. if 1, then only those
soil types will use the global value, that not have a local psi_FC set.
```

psi_FC_global = <psi in hPa> this defines the suction pressure in the soil for which WaSiM assumes field capacity. Based on this value, the percentage of usable field capacity is calculated for grids and statistics.

UsePsiFCglobalForUnassignedOnly = <[0|1]> defines, if the global value should be used for all soiltypes (0) or if the global value should only be used for such soil types that do not have any local psi_FC parameter set. If a soil type has no local psi_FC parameter, the global value is used for all horizons of that soil type.

Figure 2.14.1 shows the annual time series of the percetage of the field capacity for two sub basins of the river Inn. After hevy rainfall, the percentage can actually reach 100 % but will go down fast as the soil drains quite quickly within a few days. The uFC values in figure 2.14.1 where computed with a psi_FC_global value of 110 hPa.

As can be seen in figure 2.14.2, the percentage of the usable field capacity for the same subbasin changes dramatically by changing the psi_FC_global parameter from 110 hPa to 63 hPa. Higer values of the threshold suction pressure correspond with higher percentages – but because the maximum field capacity value for the higher suction is much lower, the higher percentage is misleading. Figure 2.14.3 shows, that the absolute value of plant available water in mm can be at the same level or be even smaller for a higher suction pressure threshold. It’s important to keep that fact in mind when evaluating percentages of the usable filed capacity.

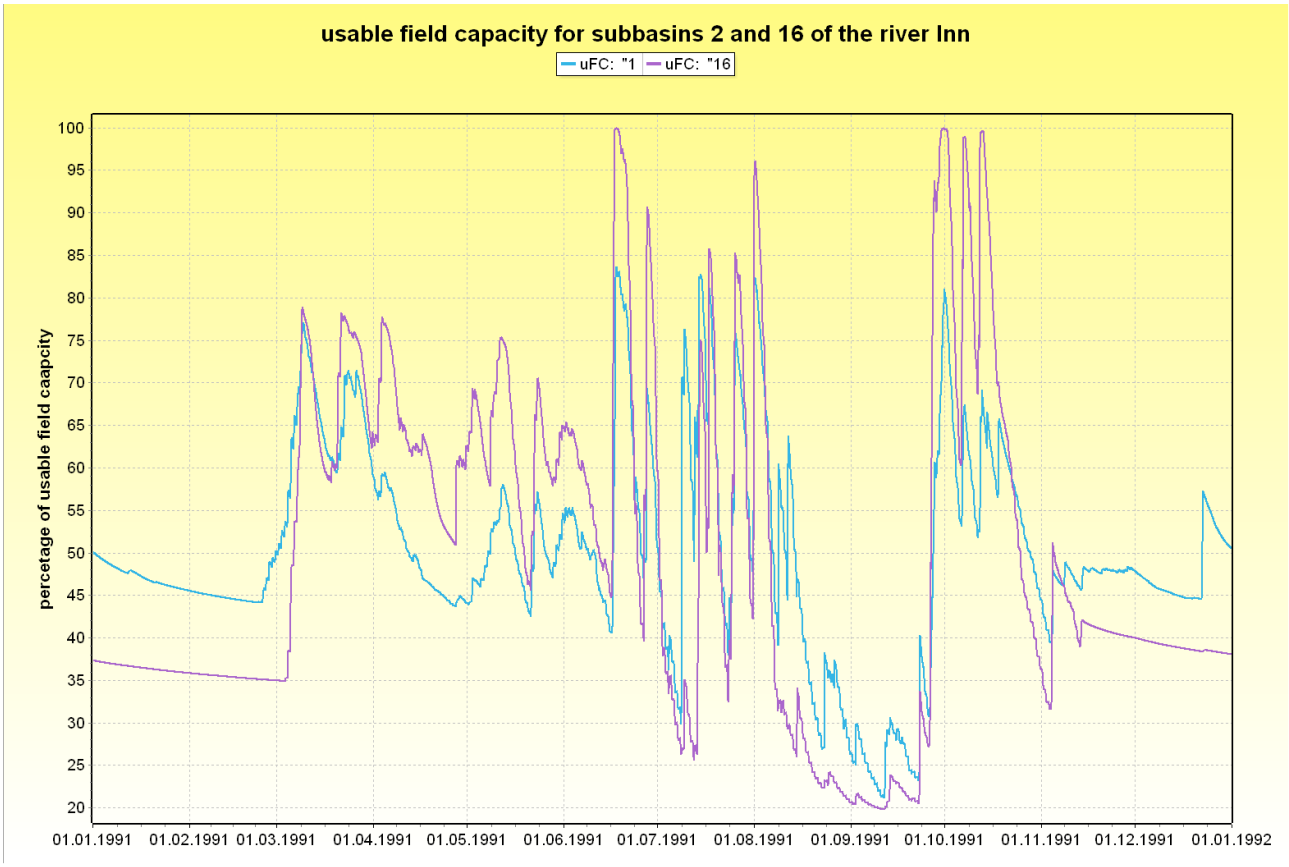


Abbildung 2.14.1: percentage of usable field capacity for two sub-basins of the river Inn

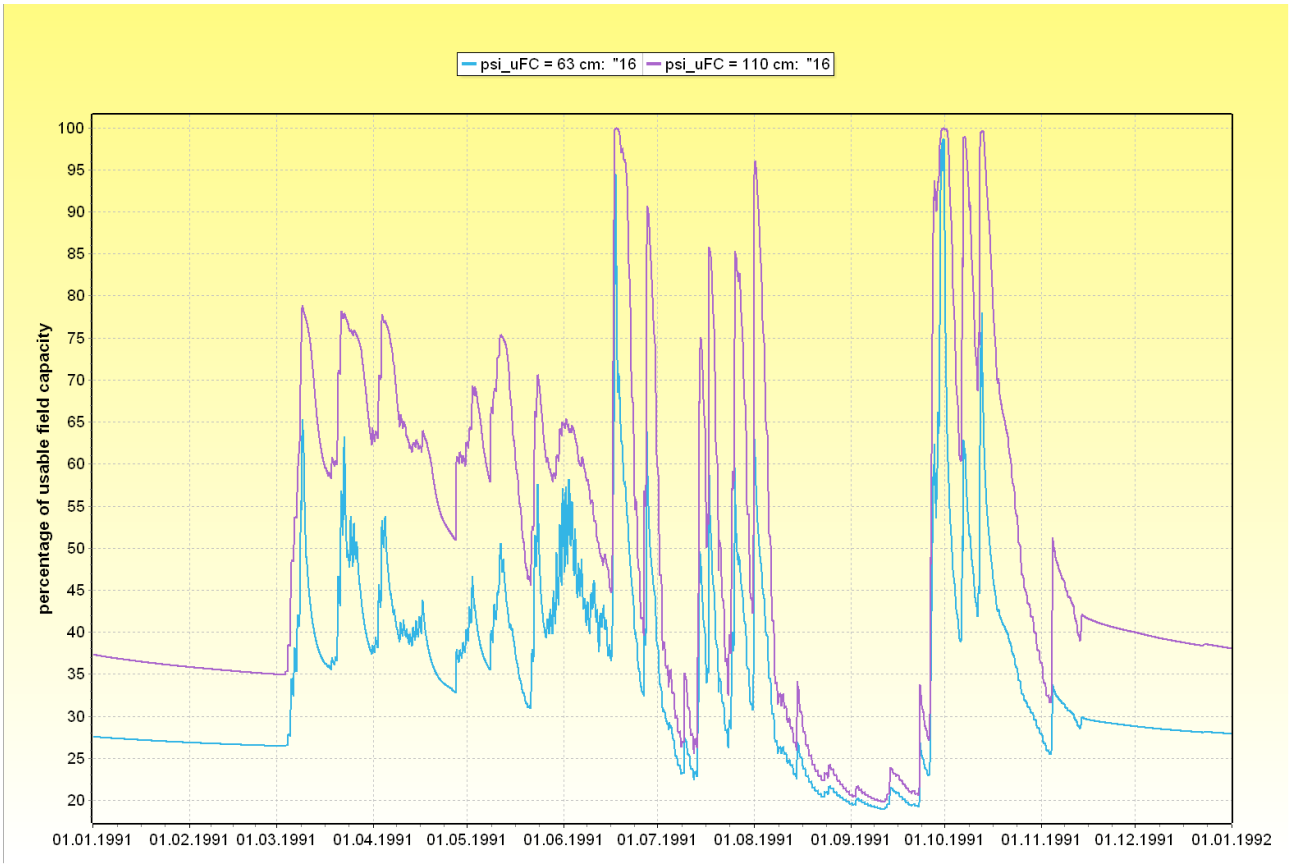


Abbildung 2.14.2: Percentage of usable field capacity for subbasin 16 of the river Inn for different suction pressure thresholds

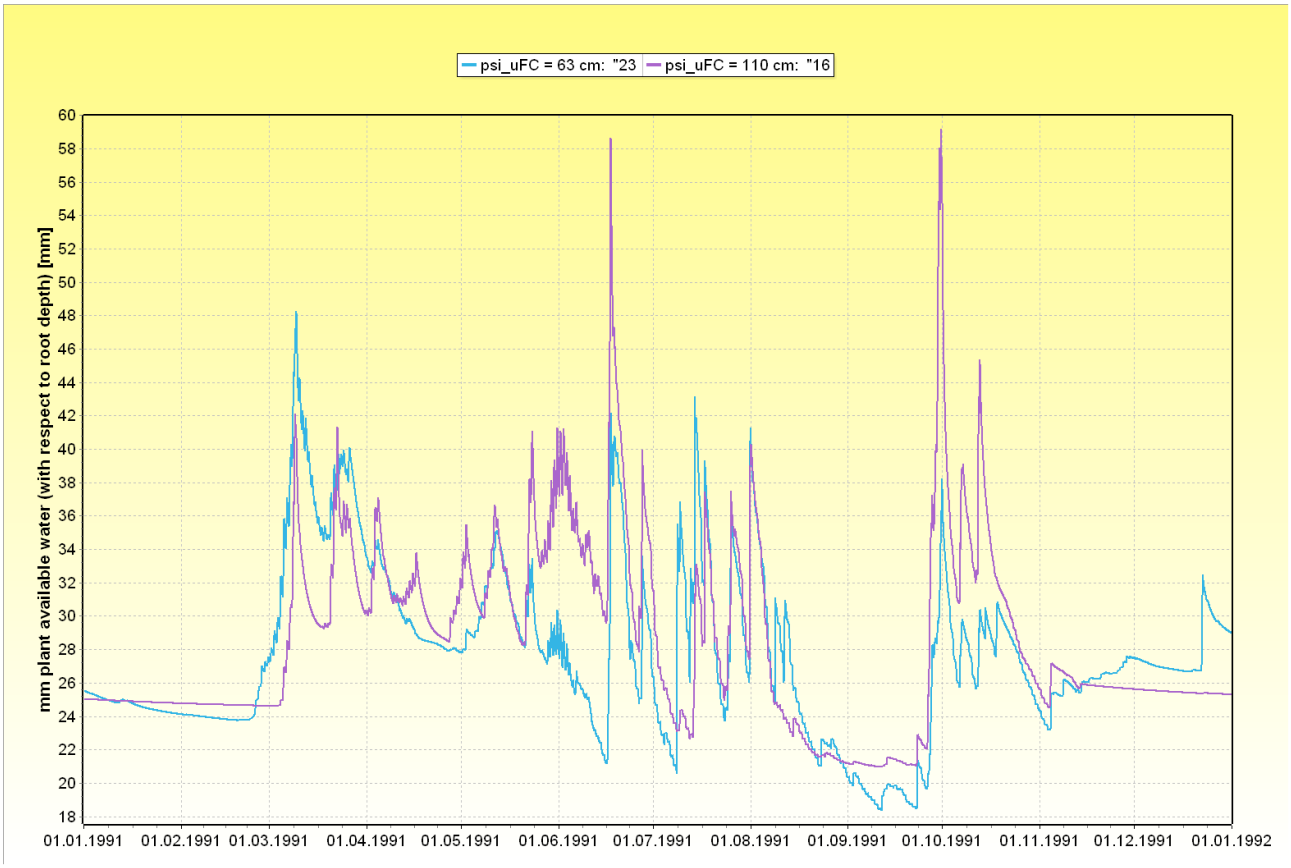


Abbildung 2.14.3: Plant available water for sub-basin 16 of the river Inn for two different thresholds (i.e. parameter psi_FC_global), blue: 63 hPa, purple: 110 cm

2.14.18 Optional Output of Soil Moisture Within Root Zone as Relative Value

The soil model of the WaSiM Richards version does already know two soil moisture grids:

```
$outpath//sb05//$grid//.//$code//$year $hour_mean # results rel. soil moisture↵
within the root zone per zone
$outpath//sb1_//$grid//.//$code//$year $hour_mean # results rel. soil moisture↵
within the unsat. zone with respect to the total volume (0m..GW table) per zone
```

The newly added option (starting in version 10.08.00) is a soil moisture grid, that stores the relative soil moisture with respect to the total pore volume, i.e. the maximum value is 1.0, the minimum value is 0.0 (or usually a bit more because there will almost always be some residual water left in the soil). Like the other newly added optional grids, the relative soil moisture grid can be defined in the section [standard_grids] and it is also possible to write a statistics file.

```
[standard_grids]
…
…
$inpath_grid//$SB_1_REL_grid SB_1_REL_grid fillcode = 1  defaultValue = 1.0 ↵
readcode = 0 outname = $outpath//$SB_1_REL_grid  writecode = 83 statfile = ↵
$outpath//sb_1_rel//$grid//.//$code//$year statcode = $hour_mean
…
```

If the unsaturated zone model finds a grid with the identification "SB_1_REL_grid", it will compute the relative soil moisture with respect to the total pore volume (i.e. $\Theta_{act}/\Theta_{sat}$) over the entire root

depth. This data may be better comparable to neighbouring basins than absolute values.

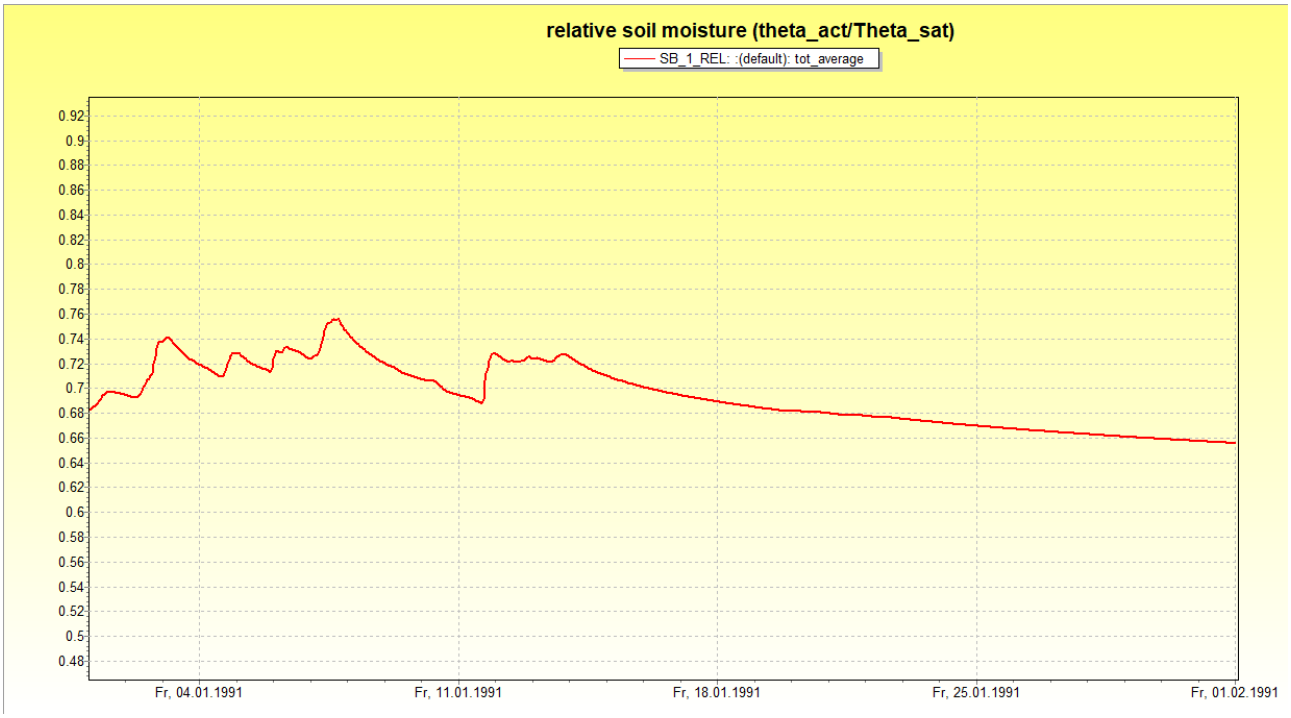


figure 2.14.18.1: statistics for relative soil moisture for the river Thur basin, January 1991

Figure 2.14.18.1 shows the relative soil moisture with respect to saturation water content (over root depth) for the Month of January 1991 for the basin of the river Thur in Switzerland.

Figures 2.14.18.2 and 2.14.18.3 show the spatial distribution of the relative soil moisture according to the graph in figure 2.14.18.1 for January 7th and January 31st. The colour scale is identical for both maps. It can easily be seen that large areas are more dry at the end of January but also that the variations between different soil types (which define the main pattern) looks very similar for both times.

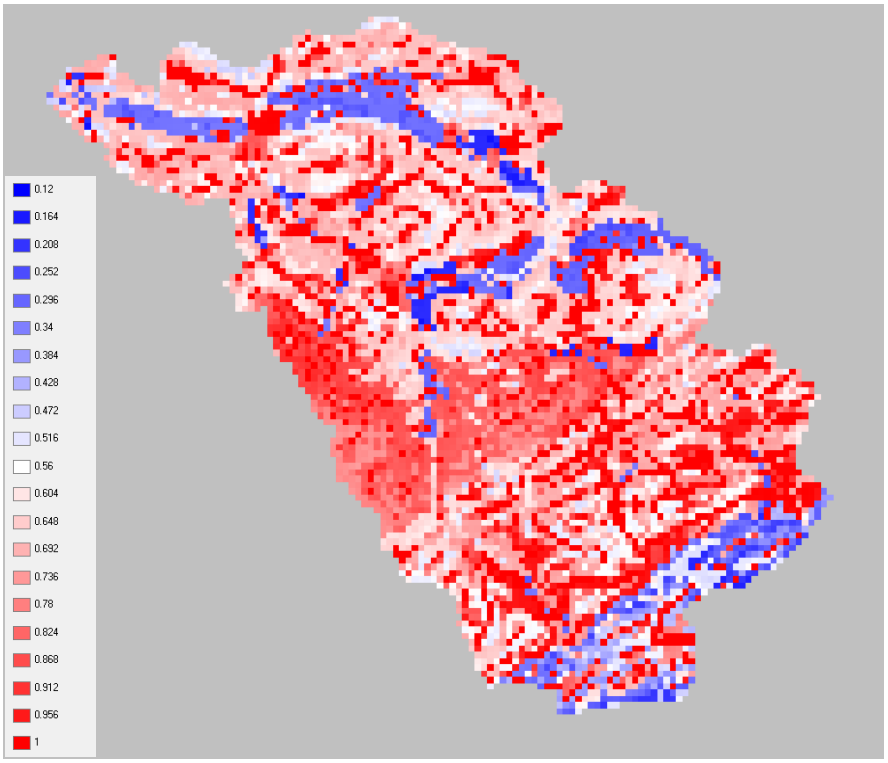


figure 2.14.18.2: spatial distribution of relative soil moisture for January, 7th, 1991, Thur basin

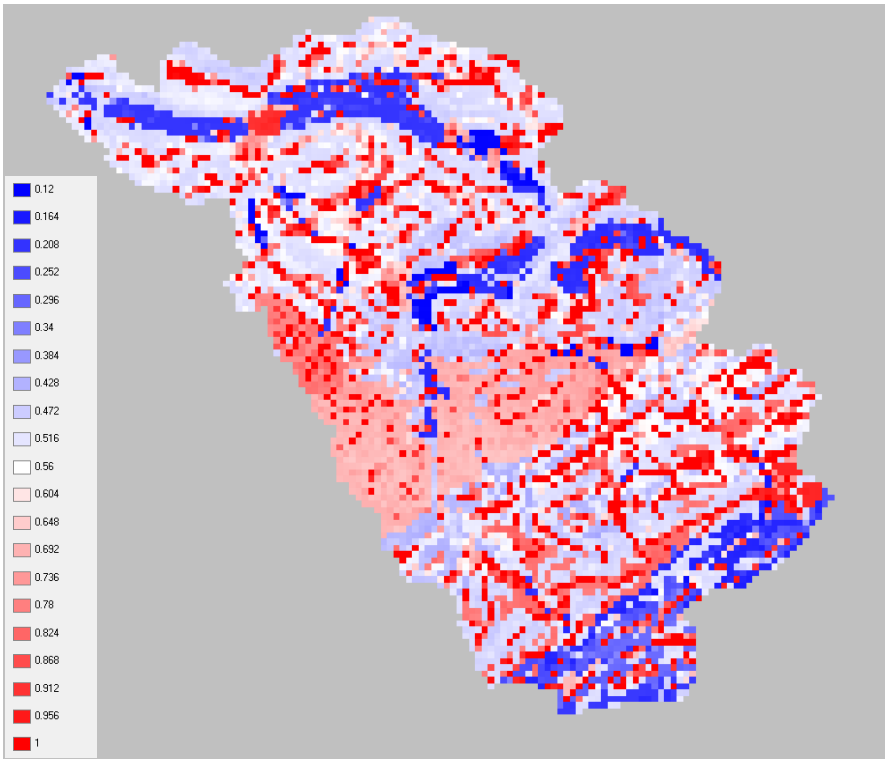


figure 2.14.18.3: spatial distribution of relative soil moisture for January, 31st, 1991, Thur basin

2.15 Modeling the groundwater flow and (lateral) transport

2.15.1 Flow equation

The main features of the groundwater module are:

- The model is a multi-layer model. Each layer is modeled horizontally 2-dimensional in a regularly spaced grid. The coupling between the layers is done using leakage factors.
- The uppermost (first) groundwater aquifer is assumed to be unconfined, the lower aquifers can be confined or unconfined (it is determined by comparing the heads with the upper boundaries of the layers)
- It is possible to use anisotropic conductivities but only in x- and y-direction (without rotated anisotropy). If a rotation is required, the coordinate system should be rotated.
- The connection to surface water is done by leakage approaches within the unsaturated zone model. It is possible to calculate infiltration from rivers as well as exfiltration into the rivers (base flow). If the groundwater reaches the surface of the soil, surface runoff is generated (but in the unsaturated zone model).
- The solution of continuity and flux equations is done using an implicit finite difference approach: Gauss-Seidel-Algorithm with automatic or manual estimation of Successive Over Relaxation Factors (SOR factors). It is scheduled to implement some faster solving algorithms like PCG into the model in later versions.
- The substance transport is done as a big-cell-approach without considering diffusion. It is possible to model the substance transport externally by writing all flux fields to external files.
- spatial distributed parameters for the groundwater model:
  - KX saturated horizontal conductivity (x-direction), 1 grid each aquifer [m/s]
  - KY saturated horizontal conductivity (y-direction), 1 grid each aquifer [m/s]
  - S0 specific storage coefficient (S_0), 1 grid for each aquifer [m^3/m^3]
  - GK leakage factors for layer interaction, 1 grid for each layer [m^{-1}]
  - BQ boundary fluxes (referring to the vertical→ recharge), 1 grid for each layer [m/s]
  - BH constant head boundary, 1 grid for each layer [m a.s.l.]
  - AQ aquifer thickness, 1 grid for each layer [m]. ***Note: Starting with version 9.08.04, the aquifer thickness provided in this grid is ignored for the uppermost aquifer and replaced by the thickness of the soil column as given by the soil table. Thus, it is made sure that the soil in the unsaturated zone model and the first aquifer are always equally thick.***

Solving the flux equation

The flux equation is get from the continuity equation and from the DARCY-equation. The change of the water storage in a control volume $\partial x \cdot \partial y$ during a time ∂t equals the balance of the inflows into and outflows out of the control volume:

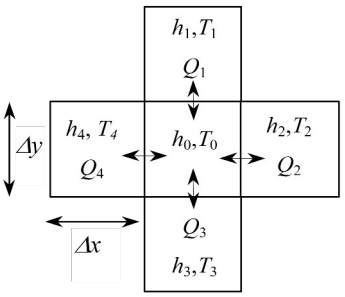
$$\operatorname{div}(T \operatorname{grad} h) + q + l_{up}(h_{up} - h) + l_{lo}(h_{lo} - h) = S_0 \frac{\partial h}{\partial t} \qquad \textbf{(2.15.1)}$$

with
- T transmissivity = aquifer thickness [m] times conductivity K_X (resp. K_Y) [m^2/s]
- h hydraulic head in the control volume[m]
- q boundary fluxes perpendicular to the grid cell surface [m/s]
- l_{up} leakage factor for fluxes between the actual and the upper aquifer [s^{-1}]
- h_{up} hydraulic head in the upper laying aquifer [m]
- l_{lo} leakage factor for fluxes between the actual and the lower aquifer [s^{-1}]
- h_{lo} hydraulic head in the lower laying aquifer [m]
- S_0 specific storage coefficient [1/1]

t time [s]

Beside the horizontal fluxes equation (2.15.1) contains also terms for fluxes between aquifers and the term q, which is introduced to consider extractions or additional inputs by wells or recharge. In case of unconfined conditions in an aquifer below the top aquifer the storage coefficient S_0 of this lower layer is replaced automatically with the storage coefficient from the top layer. This presumes that the other layers use confined values for the storage coefficient whereas for the top aquifer the effective porosity is given as storage coefficient.

To get a discrete equation, a control volume with discrete cell edges Δx and Δy is defined. Also the time is made discrete to Δt. The following equations use local indices between 0 and 4 according to the following scheme.



h_1, T_1	head and transmissivity within cell 1
Q_1	flux between cell 0 and cell 1
h_2, T_2	head and transmissivity within cell 2
Q_2	flux between cell 0 and cell 2
h_3, T_3	head and transmissivity within cell 3
Q_3	flux between cell 0 and cell 3
h_4, T_4	head and transmissivity within cell 4
Q_4	flux between cell 0 and cell 4
$\Delta x, \Delta y$	cell size of the grid in x- and y-direction

If in equation (2.15.1) the leakage term is merged together with the perpendicular inflows and outflows (including leakage) through the cells surface to the flux Q_0 and if the divergence of the fluxes is written as balance of the fluxes through the borders of the control volume than a discrete flux equation is given by:

$$\Delta t\left(Q_1+Q_2+Q_3+Q_4+Q_0\right)=\left[h_0(t+\Delta t)-h_0(t)\right]S_0\Delta x\Delta y \qquad \textbf{(2.15.2)}$$

with fluxes Q_1 to Q_4 as

$$Q_1=\Delta x T_{1,0}\frac{h_1(t')-h_0(t')}{\Delta y} \quad Q_2=\Delta y T_{2,0}\frac{h_2(t')-h_0(t')}{\Delta x}$$
$$Q_3=\Delta x T_{3,0}\frac{h_3(t')-h_0(t')}{\Delta y} \quad Q_4=\Delta y T_{4,0}\frac{h_4(t')-h_0(t')}{\Delta x} \qquad \textbf{(2.15.3)}$$

It is assumed that the fluxes during the time step Δt can be represented by the fluxes at the time t' (with $t \le t' \le t+\Delta t$) and thus by the heads valid at this time. The transmissivities $T_{1,0}$ to $T_{4,0}$ between the cells 0 and 1, 0 and 2, 0 and 3, and 0 and 4 are averaged as harmonic average of the affected cells:

$$T_{i,0}=2\frac{T_i T_0}{T_i+T_0} \qquad \textbf{(2.15.4)}$$

with i local indices between 1 and 4

Under confined conditions the local transmissivity in a cell is the product of aquifer thickness and saturated hydraulic conductivity in the matching direction. X- and y-direction may have different conductivities. Under unconfined conditions the local transmissivity is the product of the effective aquifer thickness (only the saturated part of the aquifer) and the saturated hydraulic conductivity in the direction of interest.

The following reformulations presume a regular grid with $\Delta x = \Delta y$ what leads to simplifications of equation (2.15.3) . Also the hydraulic heads h' at time t' are expressed by $h(t)$ and $h(t+\Delta t)$:

$$h(t') = (1-\alpha)\cdot h(t) + \alpha \cdot h(t+\Delta t) \qquad \textbf{(2.15.5)}$$

The parameter α has a range of 0 to 1. If α is set to 0 the solution is done explicitly, if α is set 1 the solution is fully implicit, values between 0 and 1 will cause a partly implicit solution. Explicit solutions are fast because the solution is get in a single step but they tend to numerical instabilities if using long time steps. Implicit solutions can be obtained by using iterative methods or by direct solutions of the linear equation systems. In WaSiM an iterative scheme, the Gauss-Seidel-Algorithm is applied. It is simple and fast enough also for very large model grids, because the time critical part in the model is the unsaturated zone. Usually the model will be applied with time steps of one hour or one day or even longer time steps, thus the use of the fully implicit approach is strongly recommended. If there are oscillations between successive time steps the model gives messages about this in order to allow the user to change some iteration parameters.

If the fluxes from equation (2.15.3) are simplified by $\Delta x/\Delta y = 1.0$ and put into equation (2.15.2) and if the heads $h_0(t')$ are expressed by equation (2.15.5) the head $h_0(t+\Delta t)$ can be expressed by:

$$h_0(t+\Delta t) = \frac{\frac{\Delta x \Delta y}{\Delta t} S_0 \cdot h_0(t) + (F_1 + F_2 + F_3 + F_4) - (1-\alpha)\cdot h_0(t)\cdot(T_{1,0}+T_{2,0}+T_{3,0}+T_{4,0})}{\frac{\Delta x \Delta y}{\Delta t} S_0 + \alpha \cdot (T_{1,0}+T_{2,0}+T_{3,0}+T_{4,0})} \qquad \textbf{(2.15.6)}$$

with

$$F_1 = T_{1,0} h_1(t') \quad F_2 = T_{2,0} h_2(t') \quad F_3 = T_{3,0} h_3(t') \quad F_4 = T_{4,0} h_4(t') \qquad \textbf{(2.15.7)}$$

Because after a complete step over all grid cells the heads $h_1(t')$ to $h_4(t')$ of the neighboring cells 1 to 4 may have changed, equation (2.15.7) has to be repeatedly processed for the entire grid until the changes between two iteration steps are nowhere greater than a user-specified threshold ε_{max} (of e.g. 10^{-5} m) or until the maximum number of iteration steps n_{iter} has been reached. Both parameters are specified as global parameters in the control file (global means: they are valid for each cell in the grid). To avoid to many iteration steps, for confined conditions the error threshold ε_{max} is automatically increased by a factor 50. This is done in order to consider the much larger effect of very small in- and outflows on the heads compared to unconfined conditions. Thus, a very small flux which is really unimportant for the balance may very well change the heads by e.g. 0.001 m under confined conditions whereas the same flux would change the head under unconfined conditions by e.g. 1e-6 m. Using the Gauss-Seidel iteration scheme it is taken advantage from the fact, that in each iteration step the heads of the cells 1 and 3 (local indices) are already newly calculated values which gives better estimations and thus a faster convergence of the solution.

Acceleration of the convergence of the iteration

In order to minimize the errors of the iteration even faster it is possible to use acceleration factors, the so called successive over relaxation (SOR). Using this method, the differences in the heads between to iteration steps are multiplied by a SOR-factor. This factor can be specified in the control file or it can be estimated automatically during the model run by indicating a negative value for the SOR-factor in the control file. If it is not automatically estimated, the value should be set with care, because too large values will destabilize the iteration scheme. Manually selected SOR-values should be between 1.1 and 1.4. If the model starts to oscillate, this is messaged to the user in order to change also the SOR-factor. In case of an automatic estimation of the SOR-factor the model uses the heads of the actual and the two last iteration steps to calculate an SOR-factor:

$$SOR = 1 + \frac{h(t) - h(t - \Delta t)}{h(t - \Delta t) - h(t - 2\Delta t)} \quad \textbf{(2.15.8)}$$

with SOR acceleration factor (successive over relaxation), limited to $1 \leq SOR \leq 2$ [-]

Leakage between aquifers

Fluxes between different aquifers are considered by a leakage-approach. Using DARCY's Law the flux is calculated according to the hydraulic conductivity and the thickness of the assumed clay layer in-between the aquifers. Because usually both variables are unknown they are replaced by a leakage factor l:

$$l = \frac{k'}{d'} \quad \textbf{(2.15.9)}$$

with l leakage-factor [s^{-1}]
k' hydraulic conductivity within the clay layer between the aquifers [$m \cdot s^{-1}$]
d' thickness of the clay layer in-between the aquifers [m]

The leakage fluxes are calculated according to equation (2.15.1) depending on the head difference and the leakage factor. The results are fluxes in m/s perpendicular to the grid cell surface.

boundary conditions

It is possible and required to define grids containing boundary conditions. There are two grids for each aquifer, one containing constant heads and one containing constant fluxes into or out of the cells. The boundary conditions can be set separately for each grid cell but they are valid during the entire model run (except when using time variant standard grids, offering the possibility to change the boundary conditions after a certain model time, e.g. after each year or each month or even after each day). For cells with a constant head boundary equation (2.15.6) is skipped. Inputs (positive sign) or extractions (negative sign) are expected in m/s perpendicular to the cell surface. It is also possible to consider lateral boundary fluxes if they are converted considering the grid cell size into matching vertical fluxes. It is not possible to consider temporally variable boundary conditions. Also boundary conditions which are linear combinations of known heads and known fluxes are not explicitly allowed. However, if both grids (constant flux and constant head) have valid entries for a cell, then this is a kind of a combined boundary condition. Such combinations may lead to serious problems with the water balance especially if the hydraulic properties do not match the boundary conditions (too much extraction/inflow and at the same time a fixed head → problems with the balance).

Balance for the 1 st aquifer for coupling the unsaturated zone to the groundwater model

The balance of inflows and outflows across the cell boundaries (all six surfaces including ceiling and floor) is converted into an effective rate of change of the groundwater table in terms of water flux. Thus the model of the unsaturated zone gets information about the change of the groundwater table for inclusion into it's algorithms. The rate of change of the groundwater table is estimated by:

$$\Delta_{GW} = \frac{h(t - \Delta t) - h(t)}{\Delta t} S_0 \quad \textbf{(2.15.10)}$$

with Δ_{GW} rate of change of the groundwater table in terms of a vertical water flux [$m \cdot s^{-1}$]
$h(t-\Delta t)$ groundwater head in the previous time step [m]
$h(t)$ groundwater head at the end of the actual time step [m]
S_0 specific storage coefficient [-]
Δt time step [s]

Substance transport in the groundwater

The exchange of tracers between unsaturated zone and groundwater and the mixing is considered in the unsaturated zone module. The flux between the last completely unsaturated zone and the groundwater is taken as recharge (or uptake) transporting with it an amount of tracers matching the concentrations of this tracers in the place of the origin of the fluxes. For capillary rise this solute amount is the product of the upwards flow and the concentration of the solute in the groundwater.

Within the groundwater the transport of solutes is estimated using the lateral fluxes and the solute concentrations in their origin cells. The vertical fluxes are handled in analogy to the horizontal fluxes. After calculating the water fluxes, the corresponding masses of transported tracers are calculated. The balance of mass inflows and outflows with the old tracer content of the cell (all units in 10^3kg or in relative units) is divided by the new water content to get the new tracer concentration:

$$c_0(t+\Delta t) = c_0(t)\cdot\Delta x\cdot\Delta y\cdot n\cdot d + Q_1\cdot c_{1\cup 0}(t) + Q_2\cdot c_{2\cup 0}(t) + Q_3\cdot c_{3\cup 0}(t) + Q_4\cdot c_{4\cup 0}(t) + Q_{up}\cdot c_{up\cup 0}(t) + Q_{lo}\cdot c_{up\cup 0}(t) + Q_0\cdot c_0(t) \qquad \textbf{(2.15.11)}$$

with n porosity [-]

d aquifer thickness (confined aquifer) resp. effective aquifer thickness (unconfined aquifer) [m]

$c_{1\cup 0} \dots c_{4\cup 0}$ tracer concentrations in the four neighboring cells; depending on the flow direction this may be the central cell (index 0) or one of the neighboring cells (indices 1 to 4), units [10^3kg/m^3] or relative concentrations

$c_{up\cup 0}$ tracer concentrations within the actual or the upper laying aquifer, depending on the flow direction (only if there is an aquifer above, else 0), units like $c_1 ... c_4$

$c_{lo\cup 0}$ tracer concentrations within the actual or the lower laying aquifer, depending on the flow direction (only if there is an aquifer below, else 0), units like $c_1 ... c_4$

c_0 tracer concentrations in boundary inflows (unimportant for extractions because no concentration changes), units like $c_1 ... c_4$

$Q_1 \dots Q_4$ fluxes between the cells 1…4 and the actual cell (index 0) [m^3]

Q_{up}, Q_{lo} leakage fluxes, if upper (up) or lower (lo) aquifers are present [m^3]

Q_0 boundary flux [m^3]

The fluxes Q_1 to Q_4 in equation (2.15.11) are given by:

$$\begin{aligned} Q_1 &= k_{S,1}\frac{h_0-h_1}{\Delta y}\cdot\Delta x\cdot\Delta t\cdot\frac{d_0+d_1}{2} \\ Q_2 &= k_{S,2}\frac{h_0-h_2}{\Delta x}\cdot\Delta y\cdot\Delta t\cdot\frac{d_0+d_2}{2} \\ Q_3 &= k_{S,3}\frac{h_0-h_3}{\Delta y}\cdot\Delta x\cdot\Delta t\cdot\frac{d_0+d_3}{2} \\ Q_4 &= k_{S,4}\frac{h_0-h_4}{\Delta x}\cdot\Delta y\cdot\Delta t\cdot\frac{d_0+d_4}{2} \end{aligned} \qquad \textbf{(2.15.12)}$$

with $k_{S,1} \dots k_{S,4}$ saturated lateral hydraulic conductivity in the neighboring cells [m/s]

$h_0 \dots h_4$ hydraulic heads in the cells 0 to 4 [m]

$d_0 \dots d_4$ effective thickness of the aquifers in cells 0 to 4

The amount of water within the actual cell which is needed for equation (2.15.11) may be get as balance of the old water content and the inflows and outflows or it may be estimated from the

hydraulic head at the end of the time step. If the latter value differs from the first one, the mass fluxes are reduced to the latter one. Thus numerical errors caused by the use of local hydraulic conductivities can be avoided as well as the tracer balance is hold.

2.15.2 balance check when using boundary conditions

When using boundary conditions in the groundwater model, the balance of fluxes and storage changes (which is the basic principle of a groundwater model) will be disturbed. WaSiM deals with boundary conditions in two ways:

1) constant head boundaries: the cell is excluded from the Gauss-Seidel-Algorithm because the hydraulic head doesn't have to be calculated for each time step (it is constant...)
2) constant flux boundaries: the constant flux is added as additional inflow during the execution of the Gauss-Seidel-Algorithm like leakages from above or below. However, Leakages are implicitly accounted for in the balances, because those amounts of water will internally be transferred from one layer to the other layer. Constant flux boundaries on the other hand, are additional terms which must be accounted in the balance-checksum grid.

In both cases, the initial groundwater table will be taken from the unsaturated zone model. Boundary conditions are then applied on top of this table. Thus, there will be balance differences after the Gauss-Seidel-Algorithm for constant head boundaries (because the fluxes are calculated but the head remains constant) **and** for constant flux boundaries (because the additional flux will lead to an additional change in hydraulic heads which cannot be compensated by the lateral fluxes).

The mass conservation approach of the groundwater model is:

$$\%div\left(T\cdot grad\left(h\right)\right)+q+l_{up}\cdot\left(h_{up}-h\right)+l_{lo}\cdot\left(h_{lo}-h\right)=S_0\frac{\delta h}{\delta t} \qquad \textbf{(2.15.13)}$$

where
- T transmissivity = aquifer thickness [m] times conductivity K_X (resp. K_Y) [m^2/s]
- h hydraulic head in the control volume[m]
- q boundary fluxes perpendicular to the grid cell surface [m/s]
- l_{up} leakage factor for fluxes between the actual and the upper aquifer [s^{-1}]
- h_{up} hydraulic head in the upper laying aquifer [m]
- l_{lo} leakage factor for fluxes between the actual and the lower aquifer [s^{-1}]
- h_{lo} hydraulic head in the lower laying aquifer [m]
- S_0 specific storage coefficient [1/1]
- t time [s]

When solving this equation numerically (see groundwater model description), the values of h are recalculated for each time step. As described above, the boundary conditions will disturb either the left side of the above equation (constant fluxes) or the right side (by keeping the change in storage constant due to constant heads).

The groundwater balance check grid uses these deviations to identify the impact of boundary conditions on the total balance. For each cell, after each time step, the difference between the regular fluxes and the change in storage are calculated and written to a balance-grid:

The above equation in discrete form can be expressed as:

$$\Delta t\left(Q_1+Q_2+Q_3+Q_4\right)=\left[h_0\left(t+\Delta t\right)-h_0\left(t\right)\right]S_0\Delta x\Delta y \qquad \textbf{\textit{(2.15.14)}}$$

For explanation of the entities please refer to the WaSiM description (groundwater model). When the Gauss-Seidel-Algorithm is finished, both sides should be equal, so the term

$$balance = \left[h_0(t+\Delta t) - h_0(t)\right] S_0 \Delta x \Delta y - \Delta t \left(Q_1 + Q_2 + Q_3 + Q_4\right) \qquad \textbf{(2.15.15)}$$

should be near to 0 (only very small numerical uncertainties will occur). Note: constant flux boundaries are already taken into account by the new h_0-value whereas the fluxes $Q_0 ... Q_4$ are calculated based on the heads at the beginning of the interval (without accounting for additional input or output). As can be seen by the above equation, a constant head boundary will cause the left term to be equal to 0 (because $h_0(t+\Delta t) = h_0(t)$) , only the sum of fluxes remains. Thus, if the sum of all fluxes is negative (water is flowing out of the cell because constant head is higher than neighbouring cells' head), the balance will be positive and vice versa.

For constant flux boundaries, on the other hand, the term $[h_0(t+\Delta t) - h_0(t)] S_0 \Delta x \Delta y$ will be larger or smaller than the right term $\Delta t \, (Q_1 + Q_2 + Q_3 + Q_4)$ (because boundary fluxes where added to the head in the Gauss-Seidel-algorithm). Thus, additional inflows (infiltrating water = positive constant flux boundary) will lead to positive balances, additional outflows (pumping = negative constant flux boundary) will lead to a negative balance.

Both boundary conditions could be used in the same model and even in the same cell – their effects on the balance are superposed and written to the balance grid. However: The balance grid will contain the actual value only. To get the total balance for longer time periods, the mechanisms of WaSiM for calculating sum grids must be used.

The setup of the balance grid is simple: The control file must contain a new output grid and a new write grid code, which should be either 13 (sum grid and actual grid will be written at model end) or some other combination of summing up values and writing the last grid (like e.g. 23 for sum grids for every model year).

Example for the extended control file:

```
groundwater_flow]
1                          # 0=ignore the module, 1 = run the module
$time                      # duration of a time step in minutes
1                          # solving method: 1=Gauss-Seidel-iteration
1000                       # max.numberof iterations
0.000001                   # max. changes between two iterations
0.0                        # Alpha for estimation of central differences 0.5 =
#                                  Crank-Nicholson Method, 0 = fully explicite, 1 =
#                                  fully implicite
-1.20                      # factor for relaxing the iteration if using
$readgrids                 # 1=read grids for heads from disk,
2                          # number of layers
25 68                      # coordinates of a control point for all fluxes
$outpath//glog//$grid//.//$year  # name of a file containing the flows
                                                 # between of the control point
1                          # use Pond Grid
$outpath//$head1grid       # (new) grid for hydraulic heads for layer 1
$Writegrid                 # writecode for hydraulic heads for layer 1
$outpath//$flowx1grid      # (new) grid for fluxes in x direction for layer 1
$Writegrid                 # writecode for flux-x-grid in layer 1
$outpath//$flowy1grid      # (new) grid for fluxes in y direction for layer 1
$Writegrid                 # writecode for flux-y-grid in layer 1
$outpath//$GWbalance1grid # (new) grid for balance (difference of storage change
# vs. balance of fluxes -> should be 0 or the amount
# of in-/outflows by boundary conditions
13                         # writecode for balance control grid in layer 1 (should
# be at least one sum grid per year --> Code = 20 or 23 (if old grids must be
# read in)
$outpath//$head2grid       # (new) grid for hydraulic heads for layer 2
$Writegrid                 # writecode for hydraulic heads for layer 2
$outpath//$flowx2grid      # (new) grid for fluxes in x direction for layer 2
$Writegrid                 # writecode for flux-x-grid in layer 2
$outpath//$flowy2grid      # (new) grid for fluxes in y direction for layer 2
$Writegrid                 # writecode for flux-y-grid in layer 2
$outpath//$GWbalance2grid # (new) grid for balance (difference of storage change
# vs. balance of fluxes -> should be 0 or the amount of in-/outflows by boundary
# conditions
13                         # writecode for balance control grid in layer 2 (should
# be at least one sum grid per year --> Code = 20 or 23 (if old grids must be
# read in)
```

The balance grid contains balance values in mm in order to be compatible with all the other grids and statistical output. To calculate a balance over a model run, simply add the average value of this grid to the other balance terms like precipitation, runoff, evaporation, change in soil moisture, change in snow and interception storage change in lake content etc.

2.16 Discharge routing

WaSiM does the routing based on a hydraulic calculation of the of the flow velocities. Some channel characteristics are needed for describing the geometry. The approach neglects effects of inertia and diffusion. It is a cinematic wave approach using different flow velocities for different water levels in the channel. After the translation of the wave a single linear storage is applied to the routed discharge in order to consider the effects of diffusion and retardation. The approach can be seen as a three step process: In the first step the translation is done for all channels, in the second step the storage approach is applied and in the third step the discharges from different subbasins are superposed.

Translation of discharge in the channel

The flow process in the channel is seen as stationary within a time step (but not between time steps). The flow velocity depends then for a given discharge on the stationary channel properties only (slope, roughness, width, depth etc.). Each Inflow into the channel can be assigned to a specific water level and to a specific translation time for the channel link (if regular turbulent flow is assumed and no nonlinear effects from downstream occur). After the translation time, the inflow flows out of the channel as translated outflow. To avoid an "overtaking" of slower "discharge packets" by faster "discharge packets" as it may be so in the case of a fast rising water level, the channel is subdivided automatically in as much subchannels as needed to make sure that the fastest packet is not earlier at the outlet than the slowest packet which was processed one time step before the fastest packet. The number of subchannels depends on the lengths and the hydraulic properties of the channel as well as on the length of the time step. The spectrum of flow velocities is like all other secondary data, calculated during the model initialization and stored in so called flow time tables, one for each channel. The actual translation times are interpolated using the entries in this table and rounded up to the next integer number interpretable as a multiple of the time step length.

Calculating the flow time table

The flow times are calculated using the equation after Manning-Strickler:

$$v_l = M \cdot R_h^{2/3} \cdot I^{1/2} \qquad \textbf{(2.16.1)}$$

with v_l flow velocity [m·s^{-1}]
M roughness parameter [m$^{1/3}$ ·s^{-1}]
R_h hydraulic radius [m]
I slope in the direction of the flow [m·m^{-1}]

The profile of the channel is assumed to be double-rectangular. So the discharge can be mean channel discharge and flood plain discharge. The borders between both components are the vertical extensions of the side walls of the mean channel. For each discharge $Q = Q_v + Q_h$ the discharge depths T_v and T_h, the flow velocities v_v and v_h as well as the translation time t for the actual channel and the relation of flood plain to mean channel flow are calculated. The index v stands for flood plains, the index h for the mean channel.

To get a better resolution in time it is possible to subdivide the time step into a specified number of sub-intervals. Thus it is possible to consider also non-integer multiples of a time step as translation times, because also the discharge is then subdivided into the same number of sub-intervals.

The flow time table is generated by calculating flow velocities for given discharges. Because not

only the flow velocity but also the thereon dependent hydraulic radius is an unknown variable in equation (2.16.1) , both the flow velocities and the hydraulic radius' have to be calculated by iteration. The initial guess for the flow velocity is $v_0 = 1$ m·s^{-1} for each discharge Q [m^3 ·s^{-1}], which results in an initial hydraulic R_k radius after equation (2.16.1) .

$$R_h = \frac{A_h}{U_h} = \frac{A_h}{B_h + 2 \cdot T_h} = \frac{Q_h / v_h}{B_h + \frac{2 \cdot Q_h}{v_h \cdot B_h}} \qquad \textbf{(2.16.2)}$$

with Q_h/v_h channel cross section area A_h [m^2]
B_h width of the main channel [m]
T_h flow depth in the main channel [m]
U_h effective perimeter of the main channel $U_h = B_h + 2 \cdot T_h$ [m]
v_h flow velocity in the main channel [m·s^{-1}]

combined with (2.16.1) follows the iterative equation:

$$v_h = M \cdot \left(\frac{Q_h / v_h}{B_h + \frac{2 \cdot Q_h}{v_h \cdot B_h}} \right)^{\frac{2}{3}} \cdot \sqrt{I} \qquad \textbf{(2.16.3)}$$

with M Manning-parameter [m$^{1/3}$ ·s]
I slope of the channel [m·m^{-1}]

The iteration is finished if two following iteration steps show no greater difference in the flow velocity than 0.01 m·s^{-1}. The flow depth T_h is then given by:

$$T_h = \frac{A_h}{B_h} = \frac{Q_h}{v_h \cdot B_h} \qquad \textbf{(2.16.4)}$$

If there is more discharge than the capacity of the main channel may carry, the discharge is splitted up into main channel and flood plain flow. Then a twofold nested iteration is required – one for the mean channel and the flood plains, respectively, and one for both flows together in order to join their water levels. It should be noted that the hydraulic radius for the mean channel is the width of the main channel plus 2 times the full water depth, whereas the hydraulic radius of the flood plains is only the sum of the flood plains width and the flow depth there. This implies no friction at the border from the viewpoint of the flood plains but full friction from the viewpoint of the main channel. If the water level enters the flood plains and both water levels are calculated, the difference between the water tables in the main channel and in the flood plains must be minimized iteratively. This is done in the way that in each iteration step 50 % of the flow volume difference is shifted to the flow with the lower water table. Like in equation (2.16.3) after calculating the flow velocities v_v and v_h the flow depths for the floodplains and for the main channel are calculated:

$$T_h = \frac{Q_h}{v_h \cdot B_h} \quad T_v = \frac{Q_v}{v_v} \cdot B_v \qquad \textbf{(2.16.5)}$$

with T_h Flow depth in the main channel [m]
T_v Flow depth on the flood plains [m]
Q_h discharge in the main channel [m^3 ·s^{-1}]
Q_v discharge on the flood plains [m^3 ·s^{-1}]
v_h flow velocity in the main channel [m·s^{-1}]
v_h flow velocity on the flood plains [m·s^{-1}]
B_h width of the main channel [m]

B_v width of the flood plains (sum of left and right flood plain) [m]

If T_h and the sum of T_v and the maximum mean channel depth T_m differ from each other more than by 1 cm ($|T_h - (T_v + T_m)| > 0.01$ m), the discharge ΔQ to be shifted to the other flow is calculated by:

$$\begin{aligned} \Delta Q &= -\min\left(B_h v_h\left(T_h-\left(T_v+T_m\right)\right), B_v v_v\left(T_h-\left(T_v+T_m\right)\right)\right) \quad \text{for } T_h > T_v+T_m \\ \Delta Q &= \min\left(B_h v_h\left(T_v+T_m-T_h\right), B_v v_v\left(T_v+T_m-T_h\right)\right) \quad \text{for } T_h < T_v+T_m \end{aligned} \qquad \textbf{(2.16.6)}$$

Thus, the new values for Q_h and Q_v are given by:

$$\begin{aligned} Q_{h,i} &= Q_{h,i-1} + \Delta Q \\ Q_{v,i} &= Q_{v,i-1} - \Delta Q \end{aligned} \qquad \textbf{(2.16.7)}$$

with i index of the iteration step

After calculating the resulting new flow velocities v_v and v_h the new flow depths can be estimated after equation (2.16.5) . The iteration stops, if the water level on the flood plains and the water level in the main channel differ by not more than 1 cm. The flow velocity for the total flow, which is also the velocity taken for the translation time of the discharge is given by a weighted average of both flow velocities:

$$v_l = v_v \cdot \frac{Q_v}{Q_v+Q_h} + v_h \cdot \frac{Q_h}{Q_v+Q_h} \qquad \textbf{(2.16.8)}$$

with v_l effective flow velocity of mean channel flow and flood plain flow [m/s]

doing the translation itself

The calculated translation times are rounded up to the next integer multiple of a (sub) time step. Then the inflow for that (sub) time step is shifted through the channel. Because of the differences in velocities it is possible to generate overlapped flows or even spaces in-between single discharge packets. Spaces and overlappings are balanced at each other (the overlappings are used to fill the spaces), so the balance is zero at the end of a sufficiently long period.

wave flattening by retention and diffusion

After doing the translation for a channel, which can also be only a part of a link between two subbasins, the retention is considered. This is done by applying single linear reservoirs to each of the two flow components, the main channel flow and the flood plain flow. Usually, the flood plains will have larger recession constants than the main channel, so it is possible to consider the different retention behavior. The sum of both retention results is the inflow into the next routing channel.

$$Q_{\text{out}} = \left(Q_{v,i-1} \cdot e^{-\Delta t/k_v} + Q_{v,i} \cdot \left(1 - e^{-\Delta t/k_v}\right)\right) + \left(Q_{h,i-1} \cdot e^{-\Delta t/k_h} + Q_{h,i} \cdot \left(1 - e^{-\Delta t/k_h}\right)\right) \qquad \textbf{(2.16.9)}$$

with i interval number
Δt time step [h]
k_v storage coefficient flood plains [h]
k_h storage coefficient main channel [h]
Q_{outi} outflow of the channel during time step i [mm]
Q_v discharge on flood plains after eqs. (2.16.5) to (2.16.7) [mm]
Q_h discharge in the main channel after eqs. (2.16.5) to (2.16.7) [mm]

superposing discharges from different subbasins of a larger basin

Large basins are usually subdivided into hydrological subbasins. There are source basins and in-between basins. A source basin is a subbasin without any inflows from other basins (a so called headwater basin). An in-between basin is a subbasin getting at least one inflow from other source or in-between basins. However, once the routing and superpositioning of discharges for a subbasin is done, its resulting total discharge is seen in the next lower in-between basin as coming from a source basin. After the routing scheme, the runoff from the subbasin (in both in-between and source basins) itself is superposed to the total discharge routed from the tributaries. This total discharge flows into the next in-between basin, where it is handled as inflow from a source basin.

possibilities of parameter calibration

The hydraulic parameters can often not be measured exactly. Therefore, but also because of the high variance of these parameters along a river, effective parameters have to be found. Like all effective parameters they have to be calibrated. To avoid a propagation of errors down the stream network it is possible to calibrate single routing channels by using observed instead of modeled inflows. This is done simply by replacing the calculated routed discharge by the observed value of a river gauging station and it can be controlled by a single switch in the control file.

external inflows, abstractions, and reservoirs

For each subbasin any number of external or internal inflows and abstractions as well as reservoirs can be defined. External inflows are read from files which in case of modeling also tracer concentrations should contain not only the discharge rates, given in m^3/s, but also the tracer concentrations, one column per tracer. External abstractions will also be written to files. For each reservoir, an abstraction rule as a relation between reservoir content and allowed outflow must be defined, additionally, in the case of considering tracer concentrations, initial values for all tracer concentrations must be given. The abstraction water is taken from the lowest (last) reservoir in a subbasin, even if there are multiple reservoirs. If there is more than one abstraction, they are handled in the order in which they appear in the control file until there is not enough water to satisfy the abstraction demand or until all abstractions were satisfied. If there is no reservoir, abstractions are taken from the river applying the same rule of order. Also irrigation water is taken from the last reservoir or from the river. Irrigation is ranked before the abstractions.

For further information on abstraction rules see the chapter on the lake model in this manual.

Important! For considering reservoirs the initial content as well as the initial tracer concentrations for all of the at maximum 9 possible tracers have to be specified in the control file, even if no tracer/solute transport is considered. This is true also for the WaSiM version 1 (TOPMODEL approach)!

2.16.1.1 Describing the routing structure in the control file:

Beside the usual routing of modeled discharges WaSiM offers also the possibility to consider external inflows as well as artificial (or natural) abstractions. A combination of an abstraction with an inflow is then an internal abstraction/inflow called a bypass. The impact of reservoir retention can be considered by applying an abstraction rule (a volume-runoff-table for each reservoir) or by applying a single linear storage approach to uncontrolled reservoirs.

inflows, abstractions, bypasses: Inflows and abstractions may be internal or external.
External abstractions will be subtracted from the modeled discharge, they are “losses” from the sight of the model, e.g. cooling water, irrigation water for external areas, abstractions for drinking water supply, but also losses by karst-phenomenon. Abstractions are defined as part of a routing description for a subbasin (see below). For each external abstraction a file is written containing the abstracted amount in units of m^3/s as well as the tracer concentrations of all actually modeled

tracers (up to 9 tracers at the same time), each value in a column.

External inflows, on the other hand, are discharges which were not generated within the model itself, but which are nonetheless present in the reality, like waste water disposals or, more important, like natural inflows, which are not part of the model. This be happen, if only the lower part of basin should be modeled. Then, the discharges of the upper basin part have to be considered by an external inflow (read in from an external file together with all tracer concentrations, organized in columns).

Bypasses, which are combinations of (internal) abstractions and (internal) inflows, describe the artificial or natural transfer of water within the model domain outside of the routing channel system. An example are karst-phenomenons (losses in one subbasin and additional sources in another subbasin) but also artificial bypasses by hydro power stations often abstracting water from one subbasin and adding it to another subbasin. The difference between such bypasses and external inflows/abstractions is, that bypasses will not be written into or read from external files. A bypass can also not be constructed by a combination of external inflows and external abstractions, because the model connects only these inflows/abstractions which are marked with the keyword “intern”. However, it is possible during subsequent model runs to define an external abstraction in the first model run which serves as an external inflow for another subbasin in the second model run. During the second model run the external abstraction should be defined, too (because of water balance in the lower basin areas), but it’s important to use a different file name in order to avoid overriding of the abstraction file of the first model run. The advantage of such an approach is, that the inflow is well defined in the second model run and does not depend on any model parameter. Thus, it’s also possible to use observed bypass-data.

reservoirs: If reservoirs or lakes have to be considered it is recommended to use the lake model – each lake is coded with it's unique ID in the lake grid. A routing description can deal with any number of storages/lakes, which are handled as a series of storages, but it is better to subdivide such basins into series of small subbasins. Abstractions will be taken from a reservoir, if there is one, also if there are multiple abstractions, they will be taken from the last (lowest) reservoir of that subbasin. Inflows are flowing into the uppermost (first) reservoir, if there are more than one reservoirs within the subbasin. Weirs or ground bolts should not be handled by reservoirs but by calibrating the hydraulic parameters of the channel, e.g. by calibrating single linear reservoir recession parameters for mean channel and flood plains.

Routing descriptions: For computing the discharge routing, hydraulic parameters of all channels as well as information about the drainage structure of the basin are needed. It is assumed, that the rivers are flowing into each other like in a tree-structure from the smallest branches to the trunk without branching in flow direction and without natural bypasses. This makes it possible to define an identifiable drainage structure for the basin, which could be computed subsequently link by link. However, internal bypasses would destroy this scheme, so a compromise had to be found: the abstraction parts of internal bypasses are considered in the actual time step, but the inflow parts are considered in the next time step. This scheme avoids feed backs within the same time step. It is also physically reasonable to delay the inflow by one time step, because the physical realization of the bypass also needs a certain time. The drainage structure of the basin can be edited by hand using any common ASCII-editor. It is also possible to use the program TANALYS. This program generates a routing structure which is defined by the digital elevation model and by given pour-points or automatically extracted river mouths. However, TANALYS cannot handle abstractions and inflows, whether natural nor artificial ones, and it is also not able to consider reservoirs. These features has to be added by hand after the generation of the drainage structure. If the drainage structure does not match the reality then this may lead to unexpected results.

The routing can be done either by routing each tributary through a separate channel to the outlet where all routed discharges are summed up or by summing up all tributary discharges ate the begin of the routing and routing them through a single channel to the outlet. The first method is more

appropriate if the basin subdivided according to a artificial gauge network, whereas the second method should be applied if the basin was subdivided following the natural river mouths, because the tributaries are united at the beginning of the routing channel. “Tanalys” can handle both situations. A routing description may contain any number of routing channels of both types (also mixed). All routed discharges and the runoff from the subbasin area itself are superposed at the outlet of the routing channels weighted by their relative areas on the actual subbasin total area (in-between-area of the actual subbasin plus areas of all tributary rivers which have to be routed through the actual subbasin). Each routing description may consist of any number of tributaries, external and internal inflows and abstractions, and reservoirs. The structure of the routing description is explained in the following example:

```
TG 9  (AE= 484.0, AErel=1.0)
from OL 10 (kh=0.4, kv=0.4, Bh=3.5,  Bv=20.0, Th=0.5, Mh=25.0, Mv=15.0, I=0.0128, L=10408.3, AE=3.75  )
 and OL 11 (kh=0.4, kv=0.4, Bh=7.0,  Bv=50.0, Th=1.0, Mh=25.0, Mv=15.0, I=0.0066, L=10838.5, AE=90.75 )
 and OL 12 (kh=0.4, kv=0.4, Bh=10.0, Bv=60.0, Th=1.4, Mh=27.0, Mv=15.0, I=0.0084, L=36339.6, AE=81.5  )
TG 5  (AE= 262.1, AErel=1.0)
from SUMTRIB 6&7 (kh=0.4, kv=0.4, Bh=6.0, Bv=30.0, Th=1.2, Mh=25.0, Mv=15.0, I=0.0195, L=10101.2, AE=133.75)
TG 4  (AE=1086.1, AErel=1.0)
from OL 5  (kh=0.3, kv=0.4, Bh=14.0, Bv=40.0, Th=2.5, Mh=30.0, Mv=20.0, I=0.0046, L=27005.3, AE=262.1 )
 and OL 8  (kh=0.4, kv=0.4, Bh=6.0,  Bv=50.0, Th=1.0, Mh=30.0, Mv=15.0, I=0.0090, L=26206.5, AE=16.5  )
 and OL 9  (kh=0.3, kv=0.4, Bh=15.0, Bv=60.0, Th=2.5, Mh=30.0, Mv=20.0, I=0.0037, L=26673.4, AE=488.1 )
TG 2  (AE= 208.8, AErel=1.0)
from OL 3  (kh=0.4, kv=0.4, Bh=7.0,  Bv=30.0, Th=0.9, Mh=25.0, Mv=15.0, I=0.0060, L=11952.7, AE=75.25 )
TG 13 (AE=1586.7, AErel=1.0)
from OL 2  (kh=0.3, kv=0.4, Bh=14.6, Bv=30.0, Th=2.0, Mh=40.0, Mv=15.0, I=0.0024, L=841.4,   AE=208.8 )
 and OL 4  (kh=0.3, kv=1.4, Bh=14.0, Bv=40.0, Th=3.0, Mh=45.0, Mv=15.0, I=0.0019, L=35000.0, AE=1086.1)
TG 1  (AE=1703.0, AErel=1.0)
from OL 13 (kh=0.3, kv=0.7, Bh=22.0, Bv=50.0, Th=4.5, Mh=45.0, Mv=15.0, I=0.0011, L=20289.9, AE=1586.7)
```

No inflows, abstractions, bypasses and reservoirs are contained in the above example (this is shown later). The first routing description (starting with the keyword “TG”) is defines the routing of the subbasins 10, 11 and 12 into the subbasin 9, where the separately routed discharges are superpose with the runoff from the subbasin 9 itself. Then, the routing is done for the subbasins 6 and 7, which are superposed already at the beginning of the routing channel (SUMTRIB = sum of tributaries). The routed sum of subbasins 6 and 7 is superposed with the subbasin-runoff of subbasin 5. Because the subbasins 5, 6 and 7 respectively 9, 10, 11 and 12 are independent of each other, the first routing description could have been also the second and vice versa. This is true also for the routing description of subbasin 3 into subbasin 2, which could appear anywhere before the subbasin 2 is routed itself. All these descriptions are responsible for the routing of head-water watersheds (STRAHLER-order 1) into larger (sub-)catchments of Strahler-order 2. The third description in the above example routes the discharges of the already routed subbasins 5 and 9 to the outlet of subbasin 4 and superposes them with the also routed discharge of subbasin 8 and with the internal runoff of subbasin 4. Applying this scheme, the routing is done according to the tree-structure of the catchment until the outlet of the entire basin is reached. If a model domain contains multiple basins, then its also possible to rout each basin separate using the above described commands.

Explanation of the key words in routing descriptions

- TG (German “Teilgebiet”) subbasin; the subbasin for which the tributaries are routed and superposed at the subbasins outlet
- AE = xxxx real subbasin area, the real area includes the subbasin itself and all tributaries regardless if they are included in the model or not (see AErel)
- AErel=xxx relative subbasin area; Normally, the modeled area should be the entire catchment. But if a model application only considers the lower part of a basin, e.g. the lower Ganges river or the lower Mississippi, then the model will get the inflows from the upper basin parts as an external inflow. To remain consistent with the units of the modeled runoff it’s required to tell the model, how large this upper area is (because runoff is routed as specific discharge). This is done by the relative fraction of the modeled basin parts of the total basin area.

  Example:

```
TG 9  (AE= 484.0, AErel=0.8264)
from OL 10 (kh=0.4, kv=0.4, Bh=3.5,  Bv=20.0, Th=0.5, Mh=25.0, Mv=15.0, I=0.0128, L=10408.3, AE=3.75  )
```

```
 and OL 11 (kh=0.4, kv=0.4, Bh=7.0,  Bv=50.0, Th=1.0, Mh=25.0, Mv=15.0, I=0.0066, L=10838.5, AE=90.75 )
 and ZL 1 (modus = extern geb12.dat 4 5, kh=0.4, kv=0.4, Bh=10.0, Bv=60.0, Th=1.4, Mh=27.0, Mv=15.0, I=0.0084,
L=36339.6, AE=84.0 )
```

If the subbasin 12 in the above example would not be included into the model domain, then the real area of subbasin 9 remains at 484 km^2, but the modeled area for subbasin 9 is only 400 km^2. So the relative fraction of the modeled on the total basin area is 0.8264. The routed specific discharges has to be scaled by this factor in order to be comparable to observed runoffs for subbasin 9. The inflow from subbasin 12 must be read in as an external inflow. These external inflows are given in m^3/s and they need the same parameters than the regular tributaries, because they may be natural inflows (like in the example) which are routed through real channels. The Subbasin area of such external inflows must be given only for calculating an internal runoff table.

- OL xx Zone code of the tributary to be routed (according to the zone grid for the soil model)
- SUMTRIB a&b&c… Codes of tributaries a, b, c a.s.o. which are connected by "&" (no external or internal inflows!); the discharges of all listed tributaries are superposed at the beginning of the routing channel and routed in a single channel. For each routing description (i.e. for each "TG") any number of OL's, SUMTRIB's and ZL's (see below) may be defined.
- ZL xx external or internal inflows (explanations see below)
- kh, kv Storage coefficients for single linear reservoir considering retention in a channel, units in hours, kh for the mean channel, kv for the flood plains
- Bh, Bv width of the mean channel (Bh) and the flood plains (Bv) [m]
- Th depth of the mean channel [m]; the depth of the flood plain is theoretically unlimited
- Mh, Mv Manning-Strickler-roughness parameters for mean channel (Mh) and flood plains [m$^{1/3}$/s]
- I mean slope angle along the routing channel [1/1]
- L length of the routing channel [m]
- AE (real) subbasin area if the tributary [km^2]

The input of the subbasin area AE in the first row of a routing description is required in order to be able to convert inflows and outflows as well as reservoir extractions from m^3/s into mm/time step and vice versa. It's also important to make sure that the weighted superposition of the tributaries discharges works correctly. On the other hand, the input of the subbasin area of the tributaries is only of importance for the generation of discharge-flowtime-tables at the beginning of the model run. These areas should be in such a range that the expected discharges (in mm/h) are within the discharge-flowtime-tables range as given in the control file. If these areas are much too small, the flow times for floods will be too large, because the runoff-flowtime-table will have no entries for such high specific discharges following from too small basins. The same reason is true for the input of a subbasin area for external or internal inflows. The area is needed only for generating the discharge-flowtime-table. For modeled real tributaries WaSiM checks the given areas with the areas taken from the zone grid and changes the values to the latter, if differences occur. This is not possible for inflows, so here a realistic input is required. If a inflow is an observed runoff of a subbasin which is not included in the model, this area can easily be get from the DEM or from the literature. For artificial inflows the "area" should be in a reasonable relation to the other hydraulic parameters of the channel. If, for instance, the channel has a width of 3.5 m, a depth of 0.5 m, a slope of 0.01 and Manning-Strickler-value of 25 m$^{1/3}$/s, a flow velocity of 1.332 m/s would be valid for a full channel according to a discharge of 2.33 m^3/s. If the expected inflows vary between e.g. 0.2 and 2 m^3/s, the Area could be chosen between 5 and 30 km^2, thus resulting in specific discharges between 6.67 and 66.7 l/s/km^2 (at 30 km^2) respectively between 40 and 400 l/s/km^2 (at 5 km^2). If the range for the discharge-flowtime-table is given as e.g. 5 l/s/km^2 for the minimum and 1200 l/s/km^2 for the maximum considered specific discharge, then all possible inflows are routed with a velocity

which can be found within the table (what area is given is not important as long as the range of resulting specific discharges is in-between the minimum and maximum values, the flow times are identical for all areas within the proper range because the table entries are only shifted what is compensated by the conversion of discharge from m^3/s into $l/s/km^2$ using different areas).

Routing using inflows, abstractions and reservoirs:

the configuration of abstractions, inflows and reservoirs is described in detail in chapters 2.19.7 and 2.19.8 of this documentation. See also the control file example in the appendix.

2.16.2 Runoff Routing with Observed Values in m^3/s

When using observed runoff values for routing, the routing results can be improved considerably. Until now (prior to version 10.08.00) it was required to use specific discharge values (in mm per time step). The new version allows to use observed values in m^3/s as well. All that is required is to set the mode-parameter to 3 instead of 2.

```
[routing_model]
3                    # 0=ignore this module, 1 = run the module,
# 2=run the module with observed inflows in mm/time step
# 3 = use observed runoff in m3/s
$time                 # duration of a time step in minutes
...
```

However, there are several combinations of observed runoff data and output runoff data measures and not all of them are automatically recognized, so some options will still work but result in wrong R^2-Results. The following table 2.1 gives an overview over all combinations and the expectable results

Table 2.1: Combinations of observed and modelled runoff measure and it's effect on R-sqare calculation

Nr	Routing Method	Measure observed runoff	Measure modelled runoff (defined in section output_interval)	comment
1	1	mm/time-step	mm/time-step	Default (legacy) settings, still working well
2	1	mm/time-step	m^3/s	Second version of legacy settings, still working well (mm/time step is the default expected measure for observed runoff values and m^3/s as output could be defined explicitly)
3	1	m^3/s	mm/time-step	**Routing still working ok, but R^2 calculation is not correct**, since WaSiM doesn't recognizes observed runoff as m^3/s. But since that runoff is not used for routing, the routing results are still ok.
4	1	m^3/s	m^3/s	**Same as above (i.e. routing ok, R^2 wrong**; internally, the runoff is always modelled in mm/time step, so since there is no conversion of observed runoof, R-aquare values are wrong)
5	2	mm/time-step	mm/time-step	**Works well**. R^2 is ok. This is the classic (legacy) setting for using observed runoff values for routing (always to be provided in mm/time step for method 2)
6	2	mm/time-step	m^3/s	**Works well**. Even if the observed runoff is

				given in mm/time step and the output is written in m^3/s – since WaSiM knows from the well defined routing method 2 that observed runoff is given in mm/time step and can do all required conversions internally (observed runoff has always to be provided in mm/time step for method 2)
7	3	m^3/s	mm/time-step	**Works well.** The new routing option 3 was explicitly implemented for observed runoff values in m^3/s
8	3	m^3/s	m^3/s	**Works well.** The new routing option 3 was explicitly implemented for observed runoff values in m^3/s (WaSiM can

2.16.3 Running the routing module in a shorter time step than the other modules

When the main model runs in e.g. daily time steps, the routing module may run into problems because of abstraction rules that define certain rules for certain hours of the day. For example, the power generation from pumped storage plants is usually not running over the whole day but only for some hours during the morning and the evening when the electricity demand is especially high. Abstraction rules could of course be redefined for daily runoffs and rules, but a much easier way is to run the runoff module still in an hourly time step while the other modules run in a daily time step (e.g. for modeling low flow periods). This can be done with WaSiM 11.00.05 and higher.

There are only some small changes to be made in the control file, which are listed hereafter together with input data reformatting and some general considerations for parameterization.

Control file changes:
recommended variables: a new variable for the time step of the routing module $timeRouting

```
$set $time            = 1440
$set $timeRouting     = 60

$set $starthour       = 24
$set $startday        = 01
$set $startmonth      = 01

$set $endhour         = 24
$set $endday          = 31
$set $endmonth        = 12
```

When defining the routing code, 2001 is recommended for hourly output (same as for hourly time steps). If the output should be in daily time steps, the values must either be added up by using code 4024 which sums up runoff over 24 hours (e.g. from mm/h → mm/d) or be averaged by using code 2024 (i.e. from m^3/s for 24 subsequent hours to an aberaged values in m^3/s for the entire day).

```
$set $routing_code    =  2001
# 2<nnn>  : spatial mean values for all zones (subbasin) and for the entire
basin, averaged in time over <nnn> intervals (timesteps) -> when switching the
output to daily runoff values, routing_code can be either set to 2024 (output
in m3/s) or to 4024 (output in mm/d)
```

A new optional parameter MinutesToInvEndRouting is introduced. If omitted, the default is 0. This parameter is like the fourth parameter in this section [output_interval], but specifically for the runoff module. If 0, then the hour values in the comparison file with observed runoff (which is also

the input for routing methods 2 and 3 with observed runoff) must have hour values from 1 to 24. Otherwise, if the hour values range from 0 to 23, this new variable must be set to 60, or, even better, to the variable **$timeRouting** (see above).

```
[output_interval]
25              # time steps until an output to the screen is done
1               # warning level for interpolation (no station within search
radius)
1               # output of runoff results in 0=mm/time step, 1=m3/s
0             # minutes from the hour-entry in the input data files until the
end
WriteAsciiGrids = 1 # 0 single precision binary format, 1 ESRI ASCII format
WriteDoubleGrids = 0 # 0 double format binary format, 1 ESRI ASCII format
InitialStateDirectory = $InitialStateDirectory # if using this parameter, all
state grids as well as the storage_richards.ftz file will be expected in that
directory for reading
DefaultOutputDirectory = $DefaultOutputDirectory # this is the default output
directory, all output is written to unless the given filename contains an
absolute path (starting with either / or ~ for UNIX or a drive letter and :\
for Windows
MinutesToInvEndRouting = 0
```

All meteorologic input variables must be converted to daily input series.

```
[temperature]
10   # method: 1=idw 2=regress 3=idw+regress 4=thiessen 5=bilinear 6=bilinear
 gradients and residuals linarly combined, 7=bicubic spline, 8=bicubic splines
 of gradients and residuals linearly combined, 9=read grids according to the
 name in a grid list file, 10=regression from Stationdata, 11=regression and
 IDW from station data, 12 = Thiessen with given lapse rate (as single next
 line parameter)
$inpath_meteo//T_1991-2020.day AdditionalColumns=0 # file name with station
data (if method = 1, 3 or 4, else ignored)
```

Notes on meteorologic input:

- use the tool resaggr for
  - global radiation → result is in Wh/m^2. See below
  - Precipitation
  - sunshine duration (in minutes, not the relative values)
- use the tool resmean for
  - temperature
  - wind speed
  - relative air humidity
  - vapour pressure
- when aggregating sun shine duration values, the input should be e.g. minutes per hour, aggregated values are then in minutes per day. These can be konverted to relative values using the tool sonnerel - or calculate it on your own by setting the minugtes of sun shine per day in relation to the day length in minutes for each day.

Global radiation parameters should be slighly changed

```
[global_radiation]
10  # method: 1=idw 2=regress 3=idw+regress 4=thiessen 5=bilinear 6=bilinear
gradients and residuals linarly combined
$inpath_meteo//RG_1991-2020.day AdditionalColumns=0  # file name with station
```

```
data (if method = 1, 3 or 4, else ignored)
#$inpath_meteo//global//$year//.out  # file name with regression data (if
method = 2 or 3)
650 1000 150 1 200 # lower inversion [m asl], upper inversion [m asl],
tolerance [m], overlap [0/1 for true/false], clusterlimit [m]
$outpath//$radiationgrid      # name of the output grid (is also used for
deriving names of daily, monthly, yearly sums or averages)
5//$Writegrid                    # 0, if no grid-output is needed, else one of
the codes described above
1.0                           # correction faktor for results
$outpath//rad_//$grid//.//$code//$year $hour_mean  # file name for the
statistic output (statially averaged values per time step and subcatchment...)
19998                          # error value: all data in the input file
greater than this values or lesser the negative value are nodata
$IDWweight                    # weighting of the reciprocal distance for IDW
0.5                           # for interpolation method 3: relative weight of
IDW-interpolation in the result
$IDWmaxdist                   # max. distance of stations to the actual
interpolation cell
$Anisoslope                   #  slope of the mean axis of the anisotropy-
ellipsis (-90 ... +90 degree, mathem. positive)
$Anisotropie                  #  ratio of the short to the long axis of the
anisotropy-ellipsis
0                             # lower limit of interpolation results
0                             # replace value for results below the lower limit
13000                         # upper limit for interpolation results
13000                         # replace value for results with larger values
than the upper limit
$SzenUse                      # 1=use scenario data for correction, 0=dont use
scenarios
1                             # 1=add scenarios, 2=multiply scenarios,
3=percentual change
4                             # number of scenario cells
```

Notes:

- since global radiation values for daily time steps can easily be larger than 9998, the nodata indicator must be set higher. Recommended is 19998, that means, that the input data file should **also be changed in a way that all -9999 values will be replaced by -19999**.
- The upper limit and it's replacement value must be increased for the same reason. 13000 Wh/m^2/d should be a good value here

Radiation correction should be done in 24 hourly time steps rather than for the daily time step, since this will affect evapotranspiration values

```
[radiation_correction]
1                    # 0=ignore this module, 1 = run the module
$time                # duration of a time step in minutes
2                    # control parameter for radiation correction (see above)
$outpath//$Tcorrgrid # name of the grids with the corrected temperatures
5//$Writegrid        # Writegrid for corrected temperatures
3                    # factor x for temperature correction x * (-1.6 .... +1.6)
$outpath//$ExpoCorrgrid   # name of the grids with the correction factors for
the direct radiation
5//$Writegrid           # Writegrid
$outpath//$Shapegrid # name of the grids for codes 1 for theor. shadow, 0 for
theor. no shadow (day; assumed: SSD=1.0)
5//$Writegrid           # Writegrid
1                    # interval counter, after reaching this value, a new
correction is calculated (3=all 3 hours a.s.o.)
24                   # Spitting of the interval, usefull for time step=24 hours
(then: split=24, -> each hour one correction calculation)
```

Evapotranspiration

```
[evapotranspiration]
1                    # 0=ignore this module, 1 = run the module
$time                # duration of a time step in minutes
…
0.88 0.05  # Extinktion coefficient for RG-modeling (Phi and dPhi)
1654.0     # recession constant of the daily temperature amplitude
5.3  6.4  8.1  9.9  11.4  12.0  11.9  11.0  9.8  8.0  6.2  5.2  # monthly
#values of the max. daily T-amplitudes (for 0 m.a.s.l)
0.9  0.1   # part of the temperature amplitude (dt), that is added to the
#mean day-temperature
$outpath//cloud//$grid//.//$code//$year $once_per_interval  # statistic for
subcatchments (zones) of the corrected radiation
```

Notes for evapotranspiration: It may be required to increase the montjly values for adjusting the day and night temperatures with respect to the daily average and relative sunshine duration. The example above shows values that were set 2°C higher than the default values in the control file.

Snow model:

```
[snow_model]
1                    # 0=ignore this module, 1 = run the module
$time                # duration of a time step in minutes
7                    # method 1=T-index, 2=t-u-index, 3=Anderson comb.,
4=extended com.
1.0                  # transient zone for rain-snow (T0R +- this range)
1.0 #0.0            # T0R    temperature limit for rain (Grad Celsius)
1.0 #0.0            # T0     temperature limit snow melt
…
…
0.85  # 0.5 LWINcorr: correction factor for incoming long wave radiation for
fine tuning the energy balance (accouting together with LWOUTcorr for errors in
cloudiness and albedo); recommended Values: 0.8...1.2
0.9   # 0.68 # LWOUTcorr: correction factor for outgoing long wave radiation for
fine tuning the energy balance (accouting together with LWINcorr for errors in
cloudiness and albedo): recommended Values: 0.8...1.2
```

Notes for snow model:

- because of triggering/thresholding effects, the threshold for snow/rain and snow melt may vary a bit. The tendency is for slightly higher values (out of experience)
- when using energy balance methods, due to the nonlinearity of the algorithm it is recommended to re-calibrate the LWINcorr and LWOUTcorr factors

routing model:

```
[routing_model]
1                    # 0=ignore this module, 1 = run the module, 2=run the
module with observed inflows into the routing channels (from discharge files)
3=run with modelled runoff but compare to m3/s
$timeRouting # is 60 min here
```

Here, the default variable $time should be replaced by stimeRouting (which was defined above as a new variable, holding the time step for the routing model).

Sometimes it will be required to adjust the internal time step of the loop “unsaturated zone model – surface routing – groundwater model”. The reason is, that too long time steps can lead to unstable solutions of the Richards equation and of the re-infiltration algorithms, especially if the aquifers’ hydraulic conductivity is high and also the soils’ hydraulic conductivity is high. To achieve this, an optional section in the control file can be used:

```
[soil_surface_groundwater_substeps]
24 # number of sub time steps for the module group surface routing, unsaturated
zone model and groundwater model
```

```
# (and accumulation of real evapotranspiration)
# Values to start with are 1 (default), 2
# (half of the common time step), 3 etc. Please be carefull to set too high
values here since the model
# performance will go down dramatically, since unsatzonmodel and surface
routing are called each
# time!
```

The number of sub time steps can be experimentally found if the model is already calibrated for an hourly time step by setting it to values between 1 and 24 (with the latter meaning that internally the same resolution is used as in the hourly model). Usually, a value between 6 and 12 may be already lead to adequate results. The higher the value, the longer the model run will take.

These few changes will allow the model to run in daily time step with the routing model still using the hourly time step.

2.17 Routing only mode for WaSiM

2.17.1 Introduction

Some major effects on the modeled runoff comes through routing in the river network. Not only that the discharge velocity changes with the channel profile and slope, leading to different travel times for different inflows into a routing link, but also different roughness values and especially the width of flood plains can have an major impact on the resulting discharge. Even more impact on the total discharge from a basin usually comes from abstractions from rivers and reservoirs and from the managed outflow of lakes. However, to test different values for any of the mentioned parameters, the entire model must be run – which is time consuming and therefore often the calibration of the routing model is done in a few steps only.

To overcome this situation, a special routing mode was implemented in WaSiM. It will only run the runoff routing module and is thus much faster than the complete model. However, to run routing only, the model must at least have been calibrated for all the other model components and the output of such a calibration run can then be used to run the model in routing only mode to calibrate the routing. There are still some issues to keep in mind, though (like no re-infiltration and no coupling of lakes with the surrounding groundwater is possible). Here are the key features:

- There is no new executable. WaSiM itself can do the routing by using a special command line parameter
- existing control files can be used
- The statistic file with total runoff values for each subbasin is used as input
- lakes and reservoirs are modelled using the basic version (without direct interaction of lakes with their environment)

2.17.2 Requirements in detail

- routing only mode can be used by simply adding the command line parameter -mode:routing anywhere to the command line (example: `wasimmpic64.exe controlfile.txt -mode:routing`)
- The routing model still needs some input: It uses the statistics file with the total runoff from each sub-basin.
    - To make this work correctly, an output must be generated for each sub-basin and each time step as well as for the total basin. The writecode for this statistics file must

therefore be set to 2001 in the calibration runs in normal model mode

- the statistics file with the routed runoff will overwrite the original file from the calibration run, so it should be backed up before using the routing only mode.

- „Routing only“ will run with the lake model switched off; Lakes (and reservoirs) will still be modelled, but only as „black boxes“ without considering precipitation on and evaporation from the lakes surface as well as without considering groundwater exchange. Also, only the reservoir content will be written to the output files, no water level and surface area (since the bathymetric profile is not known in the routing routines without using the lake model).
- „Routing only“ is done without re-infiltration. This can lead to large r differences between the normal model run and the routing only mode, but only in such sub-basins where a large amount of water re-infiltrates and only when re-infiltration is switched on in the normal model run at all. It is planned to implement the re-infiltration in routing-only mode later (to do this, the infiltrated amount from each source tributary must be written to a statistics file, so it can be read in later again. Actually, no such output is written, only the re-infiltration amount in the actual subbasin is written to the gwin statistics file, so the source of the re-infiltrated water is currently unknown).
- the control file can basically be left untouched, but it may be appropriate to look at the following parameters:
  - section [output_interval], 1st parameter (increment of time steps until an output to the screen is done): put a 24 here when using hourly resolution in order to get some feedback on the screen (putting 25 here means, that for hourly resolution no feedback is given)
  - when using lakes: set the initial value for the lakes volume in each routing description in order to avoid a too large bias between the normal model run and the routing only model run (background: in a normal model run, the reservoirs and lakes will be initialized from the pond-storage grid when the lake model is switched on. Since routing only doesn't use the lake model, the initialization must be done via the V0-value in the control file).
    Example: for the sample results below, Lake Thuns initial volume was set to the value from the regular model runs start value

```
 and SP 1 ( file = $outpath//spv_01_sub02.//$code//$year , V0 = 6.472E9 , C0 =
0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0, dTmin = $dtmin1 )  # Thunersee.
```

  - The storage_richards.ftz files can be used for initializing the routing channels water content, but as soon as routing parameters will be changed, the initialization becomes unpredictable. So it is recommended to set the readgrids-parameter in the control file to 0, if unwanted effects or even errors occur during the routing only mode. This should not be required for Version 10.06.00 and higher, where an automatic conversion of parameters from storage_richards files with different parameters is implemented)
- The MPI version can be used for routing only, but with one rank only.
- „routing only“ does not yet work with active surface routing
- The performance depends on the complexity of the routing descriptions. The more routing links are describes and the more lakes and reservoirs and the more abstractions and inflows have to be handled, the slower the routing will be.

2.17.3 Sample results

Figure 2.17.3.1 shows a comparison between routing results for the Kander River (Canton of Berne, Switzerland), Gauge Hondrich. A zoom-in is shown in figure 2.17.3.2. Except for the minimal slower runoff in the normal model run (due to re-infiltration), the runoff values are basically identical. Larger differences can be seen between original and routing only runoffs in figures

2.17.3.3 and 2.17.3.4, however. Here, the routing only results where obtained by changing the roughness parameters for all routing channels from their original value of 25 to 10. The change was done by “replace-all” in the control file and took some seconds. The routing-only model run took 20 seconds. So including loading the results into Graphlines, after one minute only the effect of changed roughness parameters on the routed runoffs could be studied. The results are shown without observed runoffs in order to set the focus on the effect of parameter variations on the model results.

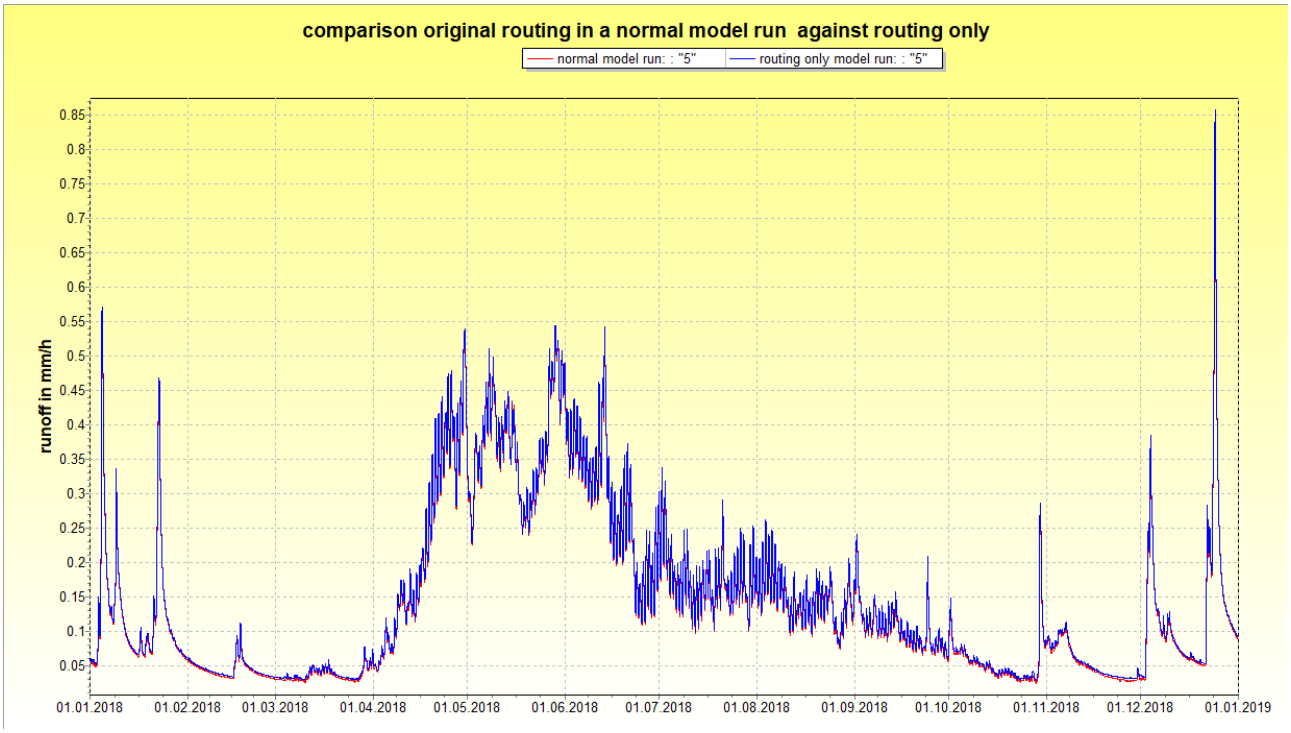


figure 2.17.3.1: Comparison between routing results for a normal model run (red) and a routing only model run (blue) with identical parameters; 2018, Gauge Kander/Hondrich (491km2)

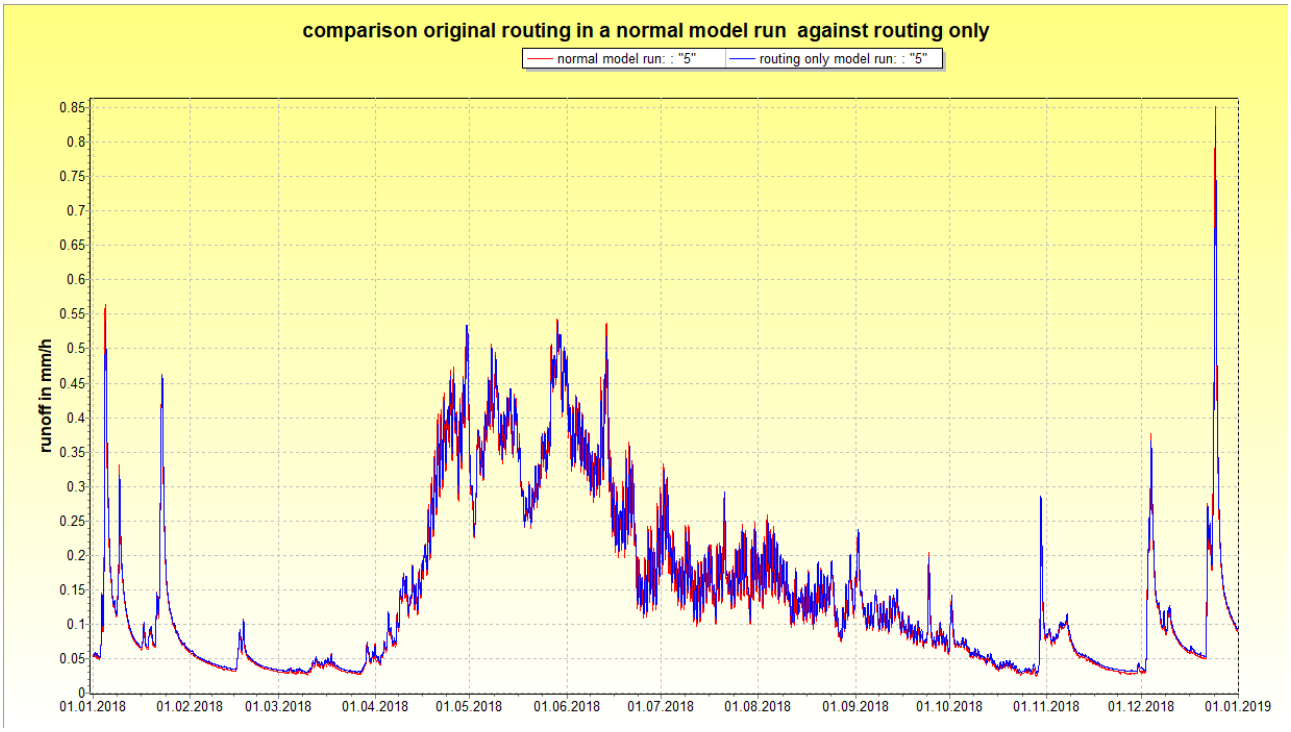


figure 2.17.3.3: Comparison between routing results for a normal model run (red) and a routing only model run (blue) with different roughness parameters; 2018, Gauge Kander/Hondrich (491km2)

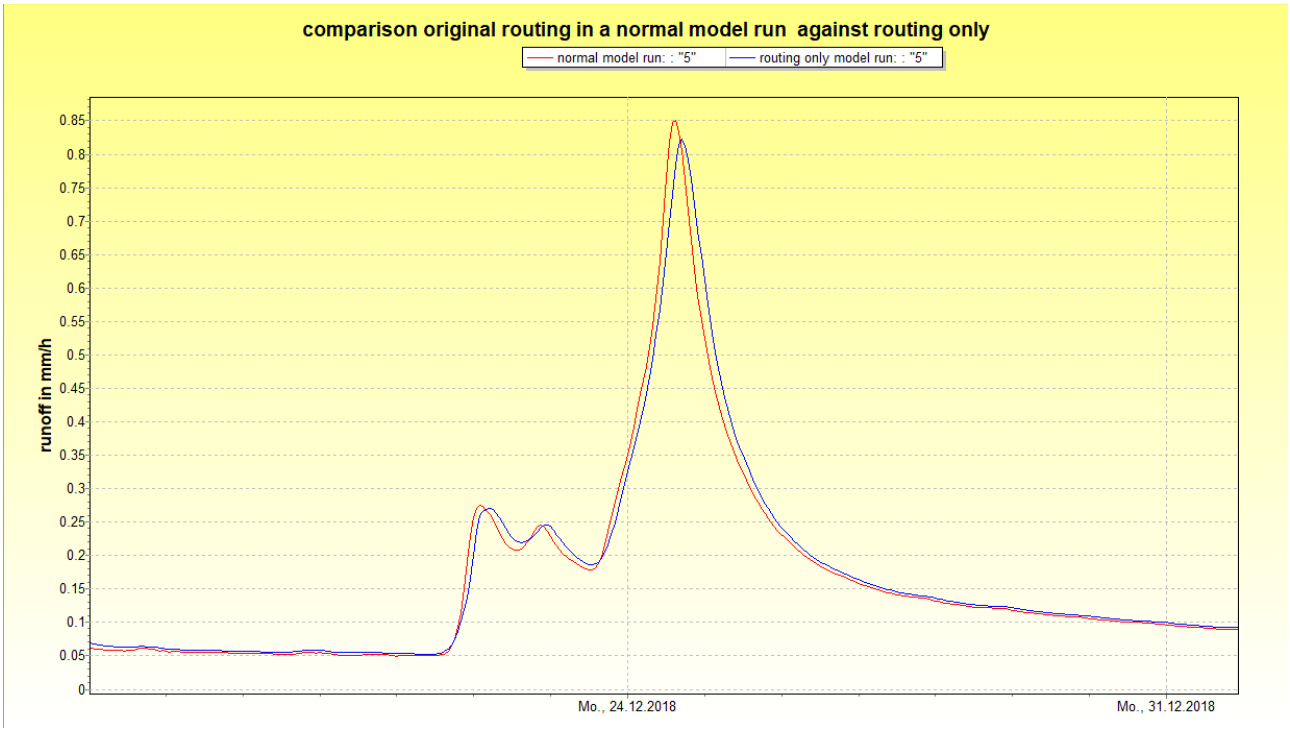


figure 2.17.3.4: Comparison between routing results for a normal model run (red) and a routing only model run (blue) with different roughness parameters; December 2018 flood event; Gauge Kander/Hondrich (491km2)

Another sample shows results for a larger basin with some big lakes in it: The river Aare upstream of Berne. Figure 2.17.3.5 shows the results (here in m^3/s from the lake statistics) for the normal model run in red and the routing model only run in blue. Here, larger differences can be identified

for peak flows, e.g. in January and December. This is due to the model considering precipitation on and evaporation from the lakes surface in normal routing, whereas routing only doesn't have these inputs (could be considered for future developments to read the lake statistics for considering precipitation and evaporation for lakes). The heavy precipitation on the lakes surface leads to an additional increase of the lakes volume, thus triggering a higher outflow in the normal model run. On the other hand, during dry periods in summer, the extra loss of water by evaporation leads to an additional decrease of the lakes volume, leading to less outflow (for instance in late June, July, August and October).

Figure 2.17.3.6 shows the corresponding content of the lake for 2018. Here, the volume of the lake reflects the different mass balancing of the integrated lake model and the routing only lake model. However, the impact is still small, so the routing only model in it's present form can be used for e.g. abstraction rule scenarios or flood protection measures scenarios.

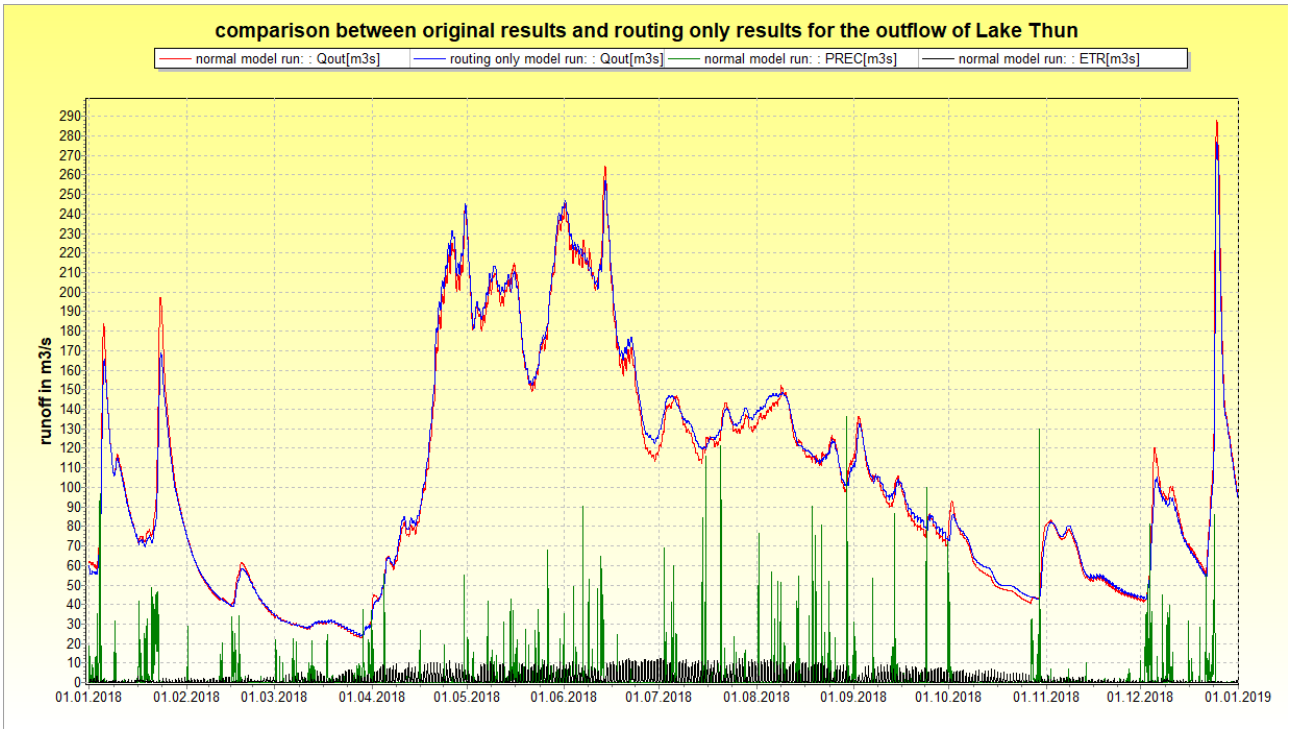


figure 2.17.3.5: conmparison between original model run and routing only model run for the outflow of Lake Thun, 2018; red=normal model run, blue=routing only model run; green=precipitation from the lake surface, black = evaporation from the lake surface

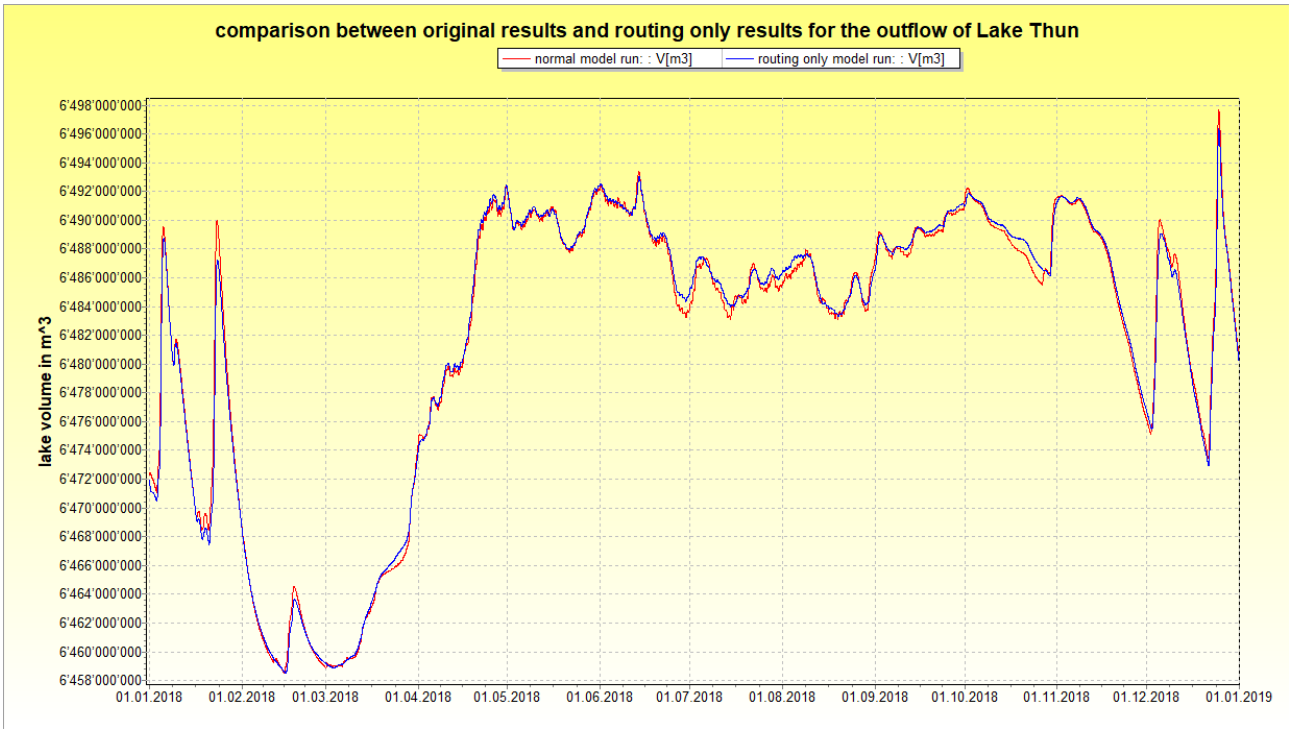


figure 2.17.3.6: Comparison between original model run and routing only model run for the content of Lake Thun, 2018; red=normal model run, blue=routing only model run

Outlook:

To improve the routing only mode of WaSiM, two enhancements are recommended:

- Considering re-infiltration. This means, that the infiltration amount for each routing channel must be saved in statistics file in the regular model run and it requires the routing mode to read in not only the total runoffs from each sub-basin but also the infiltration amounts.
- considering precipitation on and evaporation from lakes. This requires all lake-statistics to be available as input for the routing-only mode.

It should, however, be kept in mind, that both, re-infiltration and, to a lesser extend, precipitation and evaporation on lake surfaces may look different for changed routing parameters (e.g. by reducing the outflow of a lake, it's surface area may change and the re-infiltration in the downstream river links may be smaller than the input files suggest). So for a real fine tuned „finish", the regular model run can not be replaced by the routing-only mode.

2.18 Irrigation model

Modelling the irrigation is supported in several ways. There are four methods implemented:

- Irrigation Schedule 1 (code 1)
  - month/day and amount are given for a number of irrigation events during the year
    Example:

```
count MM1   DD1   amnt1   MM2   DD2   amnt2   MM3   DD3   amnt3
3     5     15    80      6     1     80      6     15    80
```

- Irrigation Schedule 2 (code 2)
  - Version 1: start date, end date, interval, and the amount are given
    Example:

```
starting from 05 01 with 20 mm to 07 31 with 20 mm every 15 days
```

- Version 2: start date, count of events, interval, and the amount are given Example:
  ```
  starting from 05 01 with 20 mm 7 times every 14 days
  ```

- Demand driven - by suction (code 3)
  - irrigation starts as soon as the suction (pore pressure) falls below a threshold (ψ_{start}).
  - The irrigation amount is calculated as the amount of water required to reach the suction at a second threshold (ψ_{stop})
- Demand driven - by ETR/ETP ratio (code 4)

Method 4 is also a demand driven method. However, the thresholds are more oriented at observable entities. The basic assumption is, that agricultural crops usually show significant drought impacts only when the real evapotranspiration rates are falling below e.g. 0.8 times the potential evapotranspiration. Irrigation thus starts at or below a given ratio of real to potential evapotranspiration. The amount of irrigation water is calculated as the difference between the actual water content and the minimal water content when no reduction applies (thus filling up the soil to a soil moisture content causing no evaporation reduction at all).

The ration of real to potential evapotranspiration is required as parameter in the control file. An entry in the irrigation table for the method 4 looks like this:

```
#Code name          method     from  minETR2ETP
19          horticulture 4                2     0.8
```

There is no upper threshold given here. Instead, the parameter HReduDry from the land use table is taken. This parameter is given as suction already.

Internally, the ratio of ETR / ETP (including transpiration and evaporation from snow and interception storage!) is calculated when the unsaturated zone model runs. The ratio is based on the total transpiration and evaporation over the entire root zone, taking into account the root density and root depth. When this ratio falls below the value given in the control file as minETR2ETP, the irrigation model will calculate the required amount of irrigation in the next time step. To do this, the actual soil water content is calculated as average over the root zone. This grid (and the accompanying statistics file, both called e.g. SB1) contains values as relative water volume (0.. Θ_{sat}). Also, in the initialization of the unsaturated zone model, a grid called SBHIGH is initialized. It contains the soil moisture for the upper threshold, which is in this case the minimum water content when no reduction of real compared to potential evaporation occurs. It's measure is relative water volume, too. The difference between this upper threshold and the SB1-grid multiplied by the roots depth is the irrigation amount which will be applied during the next time step (Δt <= 1d) or within the next day (Δt < 1d). Reductions due to river water shortage will be applied as in the other irrigation methods.

2.18.1 Optional Output of ETR2ETP Grid When no Irrigation is Active

Originally, the ETR2ETP grid was calculated for an active irrigation model only and had to be defined as variable grid in order to print the grid at all (otherwise it was only an internal grid). ETR2ETP means the relation of real evapotranspiration (ETR) to potential evapotranspiration (ETP) and should have a range of 0...1. However, by using energy balance methods, it's not impossible that there are negative real evaporation values or values that are higher than the potential value, especially for high interception evaporation rates.

The new WaSiM version now allows to put out grids as well as statistics for that value by defining ETR2ETP as a standard grid (i.e. in section [standard_grids]).

```
$set $hour_mean      =  2001
```

```
…

[standard_grids]
…                        # number of standard grids
…
$inpath_grid//$etr2etpgrid ETR2ETP fillcode = 1 defaultValue = 1.0 readcode = 0
outname = $outpath//$etr2etpgrid writecode = 83 statfile =
$outpath//etr2etp//$grid//.//$code//$year statcode = $hour_mean
…
```

The example above declares a grid with internal name ETR2ETP (so the Richards-soil model will find it) and sets the fillcode and the default value to 1 (so at least all cells are initialized with 1.0). Since the readcode is 0, no grid will be read in at startup but the grid is always created internally. The "outname" defines the output grid and the writecode defines the kind of data to be written as output. Here, every day an average grid is written at the last time step of the day. Optionally, a statistic file can be defined with an additional parameter defining the statcode (here, the variable $hour_mean was set before to 2001.

2.18.2 control file options for using the different irrigation methods

The following example is constructed to view the use of all four methods, each using other parameters as described above.

Note:

- 1 st column: this code corresponds to the irrigation code grid
- 2 nd column: the name musn't contain white spaces. It's a free-text field (use for specifying the crop type etc.)
- 3 rd column: method (0 to 4 with 0 meaning no irrigation at all, which code can be used for switching irrigation on and off for that code
- 4 th column: source of irrigation water: 1=groundwater, 2=river; can be combined with each method.
- All other columns are method specific as described in the above section.

```
[irrigation_table]
4   # number of following irrigation codes, per row one use
#
#Code name      method     from      control by
#               (0=no irr, (1=GW     demand:          table:
#               1=table1,  2=river)  psi[m]                       [mm]
[mm]         [mm]
#               2=table2)           cnt MM1 DD1 amt1 MM2 DD2 amt2 MM3 DD3 amt3
#               3=demand            start  stop
#               4=ETR/ETP<e_rp      e_rp (method 4 only)
#---- --------- --------- --------  --- ------------ ------------ ------------
 1   schedule1  1         2          3    5   15  80   6   1   80   6   15  80
 2   schedule2  2 1 starting from 05 01 with 20 mm to 07 31 with 20 mm  every
15 days
 3   demand     3         1         0.3     0.1
 4   ETP/ETR    4         2         0.8
```

2.19 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).

2.19.1 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.

2.19.2 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:

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

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

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

f) 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)

g) 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.

h) 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.

i) 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.

j) "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.

2.19.3 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:

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

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}]

- 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.

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

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

- Now, the tracer mixing of inflow and content as well as the change in concentrations due to evaporation and radioactive decay is computed.
- 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.
- 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.
- 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.

2.19.4 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, dTmin = 10, UseRunningAvg = 12 )
    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 containing 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.

2.19.5 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
```

Besides this section, a lake that should be part of the routing schema must be defined in the routing descriptions as shown above. The definition contains the following parameters:

```
   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, dTmin = 10, UseRunningAvg = 12 )
```

- reservoir ID (can be any arbitrary number, e.g. the sub basin code)
- file=<filename> (name of the statistics file)
- V0 = <initial reservoir content when starting without read in grids)
- C0 = <tracer1>...<tracer9> (initial tracer concentrations, all 9 entries should be present, even if not used)
- (optional) dTmin = <n in seconds> internal sub time step for balancing inflow and outflow, for hydraulic connections sometimes even less than a second, otherwise usually around 10 seconds to several minutes
- (optional) UseRunningAvg = <n intervals> (using the running average of the last <n> intervals to define the outflow of the reservoir. If the parameter is not provided or is less than 1, the defaoult behaviour is to use the current reservoir content

2.19.6 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.
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.

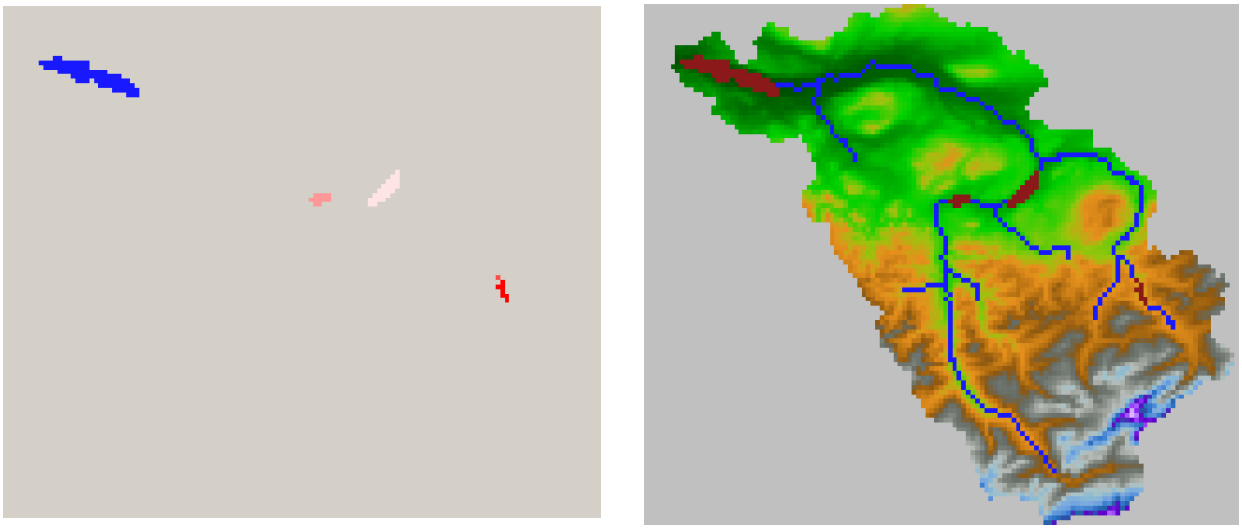


figure 2.19.6.1: Codes for lakes, here as a fictive example for the river Thur basin in Switzerland (these lakes do not exist!). The right picture is only for better imagination, the left one is the real input grid.

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).

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. Figure 2.19.6.2 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.

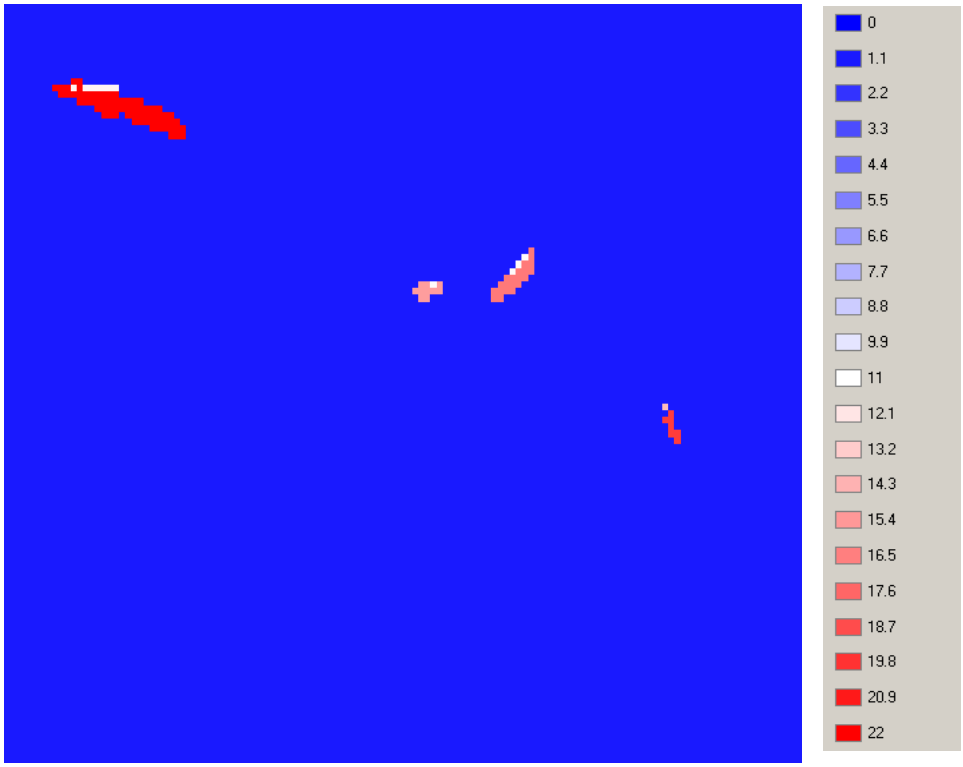


figure 2.19.6.2: MAXPOND grid for the above lake grids from figure 2.19.6.1

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.

2.19.7 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 source 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 0 with the first real data column → first 4 columns are for year, month, day and hour and do NOT count)
<outputfile> path and file name 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. This value refers to the discharge in the target sub-basin. The modelled discharge in the target sub-basin (which may have been reduced by abstractions from that target basin as well) and the sum of all inflowing abstractions should generally not be larger that the value given by TargetCap. However, since not all abstractions and inflows may be known when a certain routing description is processed it may happen that a reduction will be applied not in the actual but in the next time step only (i.e. when the target sub basin will be routed after the actual abstraction is applied, the results for the total runoff in the target basin may still be above the capacity, because there may be a rising flood event in the target basin that couldn't be considered in the source basin yet when the abstraction is calculated)

Note: If the abstraction should be limited by the capacity of the abstraction itself (e.g. because of small pipes or so), the mapping table itself needs to be designed accordingly, e.g. by adding a last row that maps an unrealistic high runoff or storage content value (e.g. 1e6 m^3/s or 10

^{9} m ^{3}) to the maximum allowed abstraction value.
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    366
7            0      0      0      0      0      0      0      0      0      0      0
8            1      0      0      0      0      0      0      0      0      0      1
9            2      1      0      0      0      0      0      0      0      0      2
10           3      2      1      0      0      0      0      0      0      1      3
11           4      3      2      1      0      0      0      0      1      2      4
14           7      6      5      4      3      3      3      3      4      5      7
15           7      7      6      5      4      4      4      4      5      6      7
16           7      7      7      6      5      5      5      5      6      7      7
17           7      7      7      7      6      6      6      6      7      7      7
18           7      7      7      7      7      7      7      7      7      7      7
27           7      7      7      7      7      7      7      7      7      7      7
27           8      7      8      7      8      7      8      7      8      7      7
TargetCap =  80     80     80     80     80     80     80     80     80     80     80
WeekDays  =  12345  12345  12345  12345  12345  12345  12345  12345  12345  12345  12345
start_hour = 6      6      6      6      6      6      6      6      6      6      6
stop_hour = 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    366
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
2.925e08     2      2      2      2      2      2      2      2      2      2      2
3.0875e08    10     8      5      3      10     8      5      3      10     8      3
3.25e08      40     30     20     10     40     30     20     10     40     30     10
3.43e08      200    150    100    70     200    150    100    70     200    150    70
TargetCap =  400    400    400    400    400    400    400    400    400    400    400
WeekDays  =  67     67     67     67     67     67     67     67     67     67     67
start_hour = 6      6      6      6      6      6      6      6      6      6      6
stop_hour = 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.

2.19.8 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

2.19.9 Abstraction Rules for hydraulic connections between lakes

B eginning with version 10.02.00, a new kind of abstraction rules was introduced. The actual problem to solve was the hydraulic connection (free water table flow) of three lakes in Switzerland (the so called Great Jura lakes Bielersee, Neuenburgersee and Murtensee). These three lakes are part of the so called “Juragewässerkorrektion”, a project for flood protection and drainage that was carried out in two major steps between 1868 and 1973. Figure 2.19.9.1 show the situation of the three lakes. The river Aare (with around 5000 km ^{2} upstream area, not shown here in the picture) flows into lake Bielersee. The normal outlet is at the weir Brugg (marked with the arrow and the word outflwo). The three lakes are connected by big channels, the lakes water tables are pretty identical, except when a flood is coming in from the Aare river, building a gradient from the bielersee to the Neuenburgersee and then from the Neuenburgersee to the Murtensee that reverts the natural runoff direction from Murtensee into Neuenburtgersee and from Neuenburgersee into Bielersee.

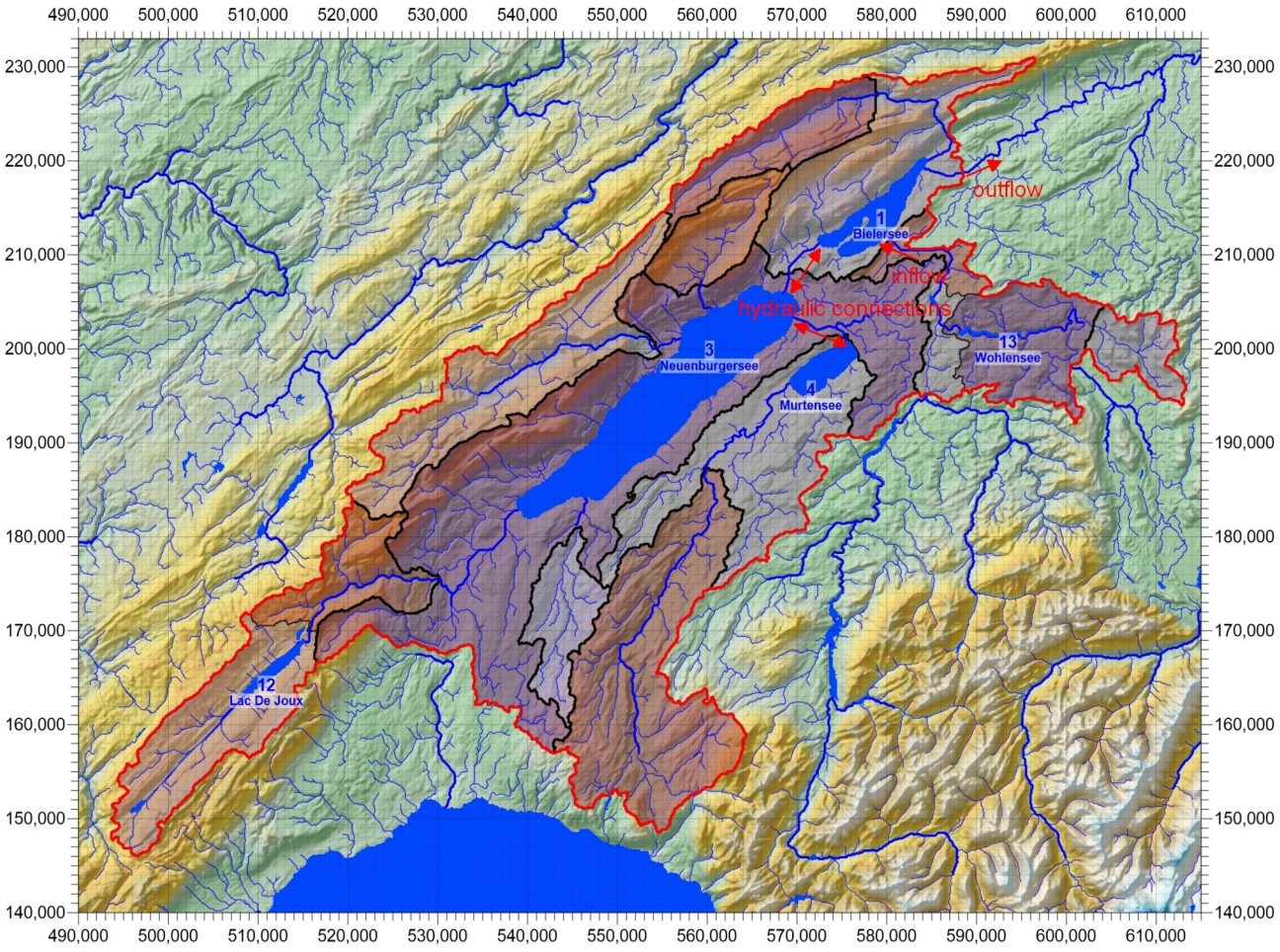


figure 2.19.9.1: example for hydraulic connections between three lakes

To model such flux reversion in WaSiM, the classic abstraction rules had to be extended by a more physically based approach that takes account on the lakes water balance, the modeled water tables and the hydraulic properties of the channels between the lakes.
The solution is no a new kind of abstraction rule called HydraulicConnection (what describes the physical background pretty good). The setup is relatively easy:

```
[abstraction_rule_reservoir_3] # Neuenburgersee
-1 # special tag to mark the following hydraulic rule
HydraulicConnection
{
```

```
        TargetReservoir = 1;
        ChannelWidth = 67;
        ChannelBottom = 423.8; //in m ü.M.
        ChannelLength = 8500; // in m
        RoughnessCoeff = 45;
}
```

There are some hydraulic parameters that describe the geometry (width, length) and the absolute height of the channel bottom as well as the roughness parameter. The lake model keeps track of the lakes water table, so together with the geometry and the bottom height, a kinematic wave approach can be applied. This does not account for dynamic effects, but is is usually robust enough for slower changes (it takes a few hours to fully develop the flow reversal).

An important parameter is of course the target reservoir. This is the code of the lake (not the sub basin) the actual sub-basin is draining into following the normal gradient of the DEM.

Figure 2.19.9.2 shows the effect of the new abstraction rule. Green is the modeled runoff of sub-basin 3 (with lake ID 3) to sub-basin 1 (lake ID 1) without hydraulic connection. Red is the observed runoff in the Zihlkanal, the channel between lakes 1 and 3. Blue is the modeled runoff with the new hydraulic connection abstraction rule.

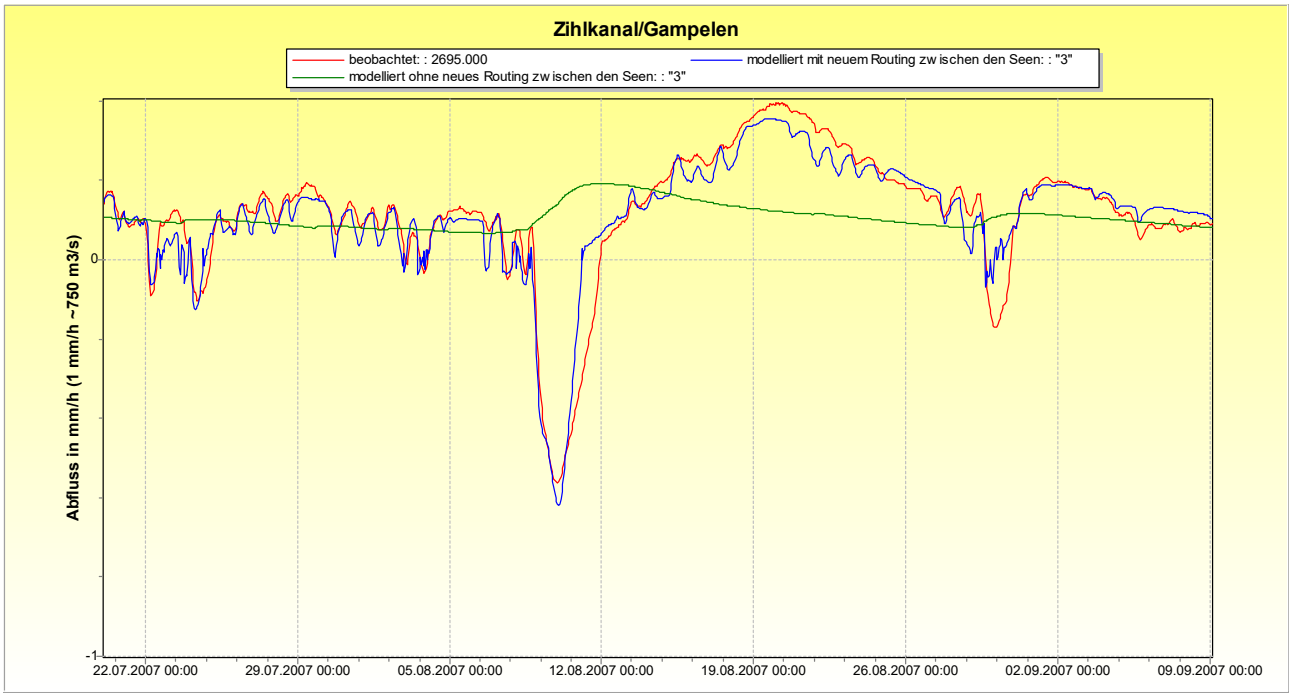


figure 2.19.9.2: example for flow reversals between lakes Bielersee and Neuenburgersee due to the new abstraction rule as hydraulic connection

Figure 2.19.9.3 shows the lake water table of lake Neuenburgersee for the year 2007. Green is again the modeled water table without and blue the modeled water table with hydraulic connection. Red is the observed water table. It can easily be seen that this new abstraction rule is a big advantage compared to the conventional one-way-routing using the normal abstraction rule.

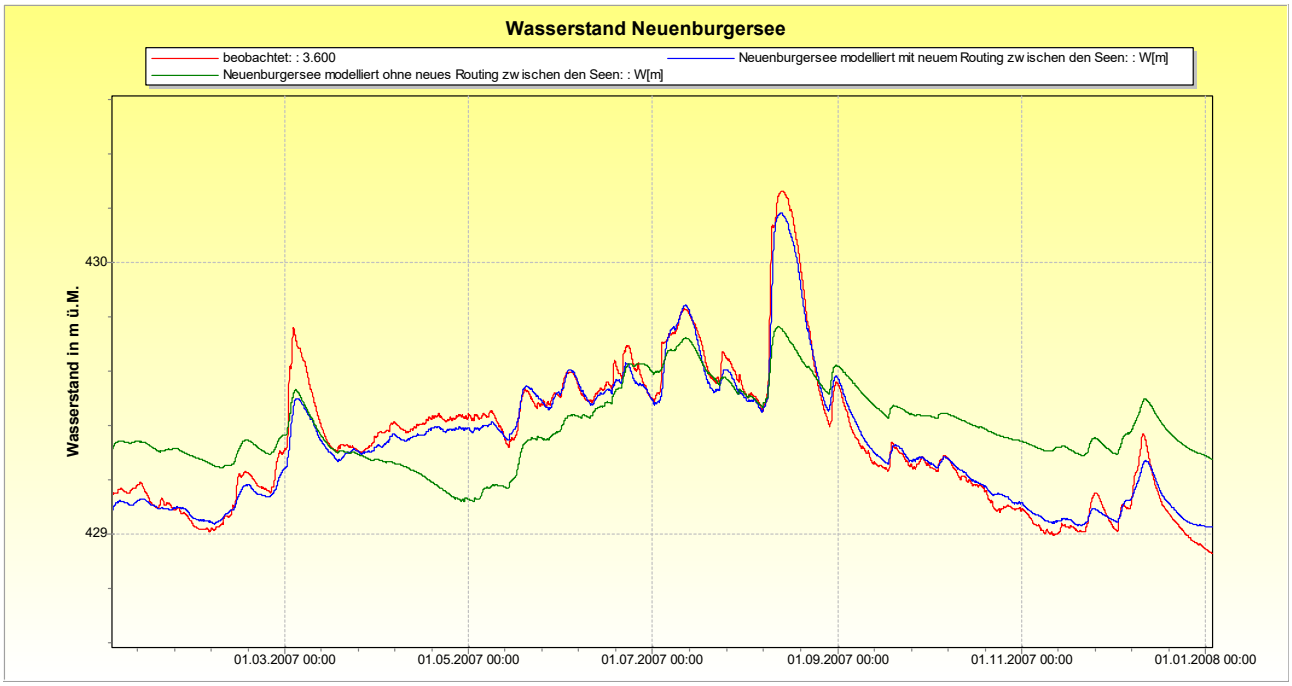


figure 2.19.9.3: Example for modeled lake water table with and without the new hydraulic connection abstraction rule (red=observed water table, green=modeled water table without hydraulic connection, i.e. classic one-way-routing, and blue=modeled water table with hydraulic connection)

2.19.10 Samples of lake modeling with time variant abstraction rules

figure 2.19.10.1 and 2.19.10.2 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.

In figure 2.19.10.2 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

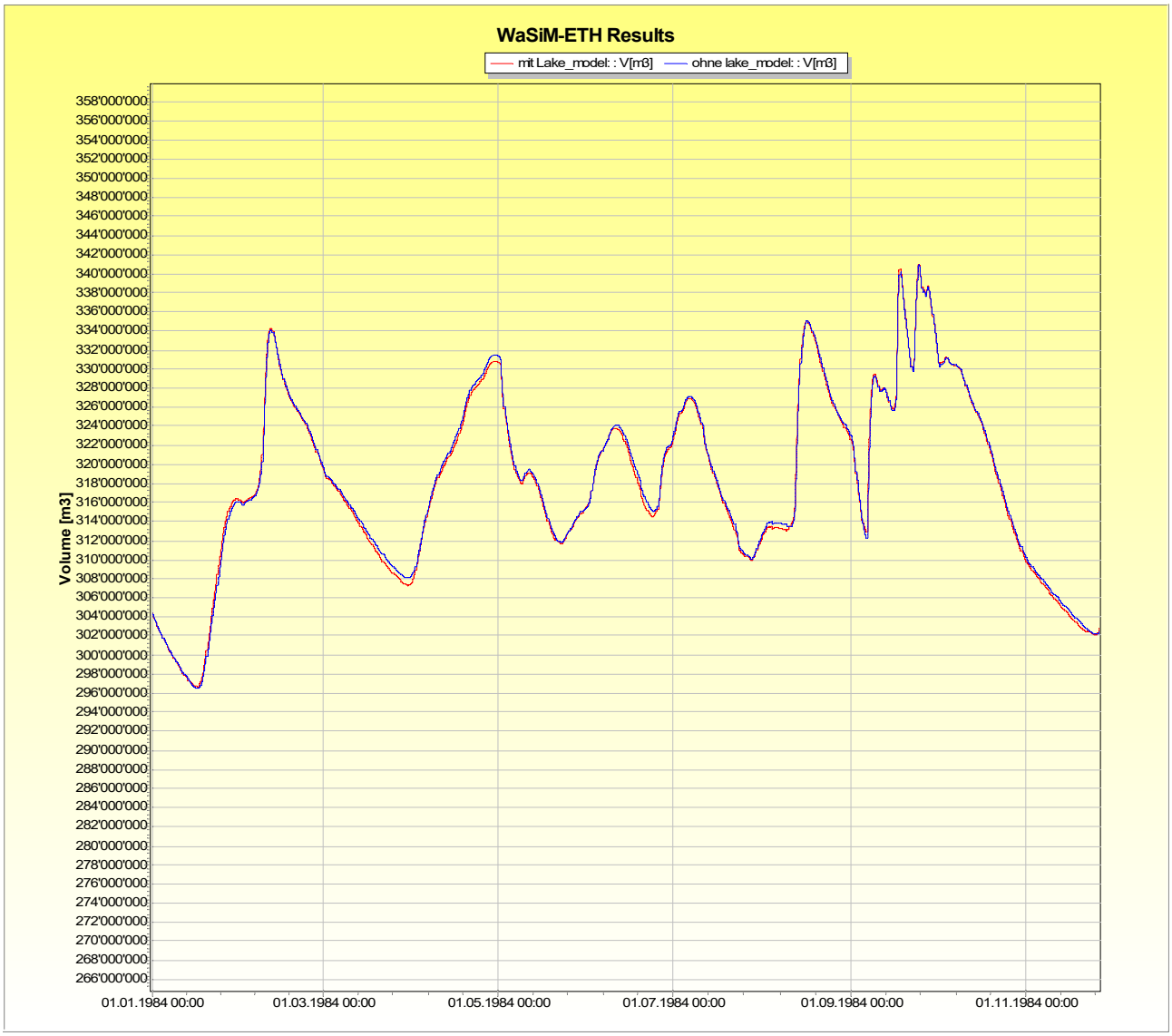


figure 2.19.10.1: comparison of lake volume with the fully integrated lake model and with the simple (routing module only) lake model

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.

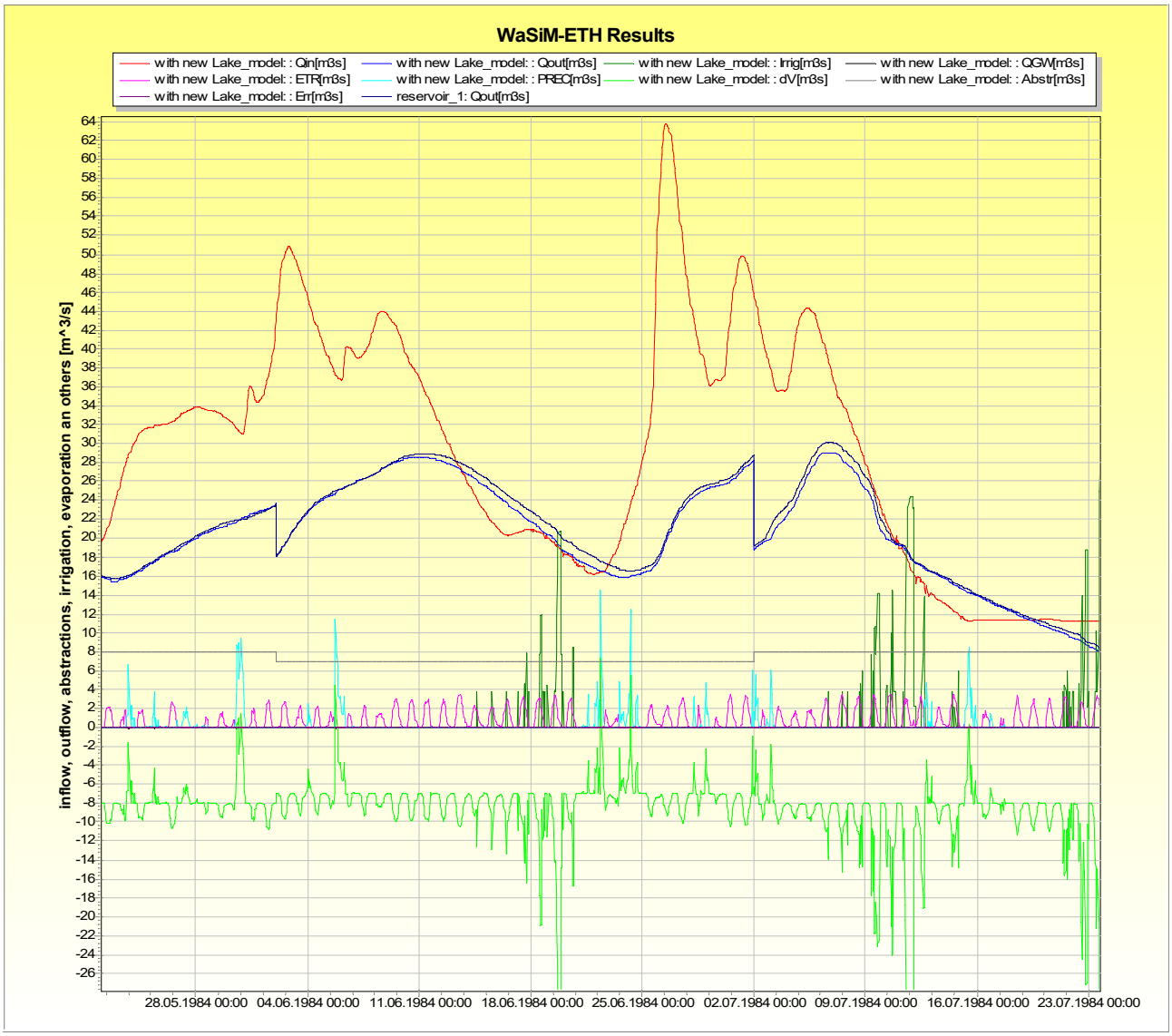


figure 2.19.10.2: some components of the lakes water balance statistics.

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

2.19.11 A new abstraction rule for lakes and reservoirs using externally provided runoff data

When computing abstractions for lakes, the rules defined for either the downstream runoff or the abstractions into other sub-basins are usually not as flexible as reality requires. For example, if a lakes or reservoirs outflow is controlled following a time dependent rule (which day within a year do we have, what is the actual reservoir content or water level), but there are external factors, the rule doesn't account for (like a requirement for letting a flood wave pass in a downstream river without adding too much runoff from the reservoir), then the outflow or abstraction will be wrong although it perfectly followed the abstraction-rule.

To overcome this problem, WaSiM 11.00.00 introduces a new type of abstraction rule, which is of type "intern_from_file_from_reservoir". That rule allows to take the abstraction values from an external file and replace the internally computed values. The rule is special in the sense that the normal abstraction rules will stay in place and are also processed, but the external abstraction replaces the internal value only if there was a valid (non-nodata value) found in the external source.

A problem that arises when using such external abstraction data is, that the balance between reservoir or lake content and abstraction is no longer given, i.e. when the model tends to overshoot or undershoot with regards to the inflow to a lake or reservoir, the lakes water level will drop or rise in unexpected ways. It is therefore highly recommended to always use the new abstraction external rule together with a water level assimilation (configured in section [data_assimilation] of the control file.

To configure the new abstraction rule, the SP routing description was extended:
Consider the following rule

```
…
 and SP 3 ( file = $outpath//spv__03_sub21.//$code//$year , V0 = 92.2E6, C0 = 0.0 ↵
0.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0, dTmin = $dtmin1 )  # Sihlsee C0: initial ↵
concentration for all of max. 9 tracers in the reservoir water; optional parameter ↵
dTmin is in seconds (60s is default even without this parameter given)
 and AL $28 (modus = intern_from_file_from_reservoir , externalAbstractionFile =
$inpath_hydro//givenAbstractionsSihlsee2.dat 1, replaced_Abstractions = { 0 $1 $2 } ) ↵
# that means, that this abstraction is read in from an external file, reading in ↵
column 5 and the values will replace the abstraction $1 and set abstraction $2 to 0
 and AL $1 (modus = intern_with_rule_from_reservoir ) # used for residual water
 and AL $2 (modus = intern_with_rule_from_reservoir ) # used for residual water
```

The new abstraction rule $28 (in red) has some new parameters:

- modus = intern_from_file_from_reservoir → defines the new abstraction rule type
- externalAbstractionFile = <filename> <datacolumn> → defines the filename to be read and the data columns (first valid column is without YY, MM , DD, HH and mm, 1-based).
- ReplacedAbstractions = <list of abstractions to replace) → a list of abstraction ID's which will be replaced by the external values.
  - 0 → this is the regular outflow of a lake or the flood overspill for reservoirs, flowing in the regular downstream sub basin)
  - any other number → an abstraction, defined in the same TG routing descrition.
  - The rule is as follows: The first listed abstraction to br replaced gets all the external abstractions, whereas ALL other abstractions to be replaced will be set to zero (0). So, the downstream outflow, which is usually defined by the [abstraction_rule_reservoir_<n>] is replaced with the externally read value and all the other abstractions, defined by [abstraction_rule_abstraction_<n>] rules, will be set to 0.

the following pages contain results from a sample application from to the lake Zurich sub-basin.

Figure 2.19.1 shows the situation “as was” before adding the external abstraction rule. Runoff and water table were modeled using the rule defined for abstraction 1 for SP 1. The corresponding water table is always near the observed water table (because the rule reduces outflow when the water table drops too much and it increases the outflow when the water table rises too much, both entities are in an equilibrium defined by the rule. Figure 2.19.2 shows a very similar picture for the water tables. As can be seen, although the runoff in the first months of the year is muss less than observed, the water table drops only a bit below the observed water tables.

Now, when applying the new abstraction rule with externally read outflow values (here, we used the observed runoff on an hourly basis), the “modeled” runoffs are as expected identical to the observed ones (firgure 2.19.3). However, the modeled water tables differ dramatically from the observed ones, see figure 2.19.4. This is due to imbalancing outflow and water table as it was still the case when using the internal rule.

To overcome this, the assimilation of observed water table values was added in another model run. Figure 2.19.5 shows the change in water table when adding that assimilation. Here, we let the model assimilate observed water tables only once a day. The effect is, that for the time step with assimilation, the water table in the model is adjusted to the observed value but starts to drift off as soon as the model is not using the observed water table values until the next synchronization takes place. Figure 2.19.6 shows a zoom-in for some months, so, the effect can easier be seen.

The examples show, that external abstractions can be used for lakes but should always be accompanied by water level assimilations, if possible with the same time step than the model itself is using (even a single day leads to larger differences in observed and modeled water table vaules.)

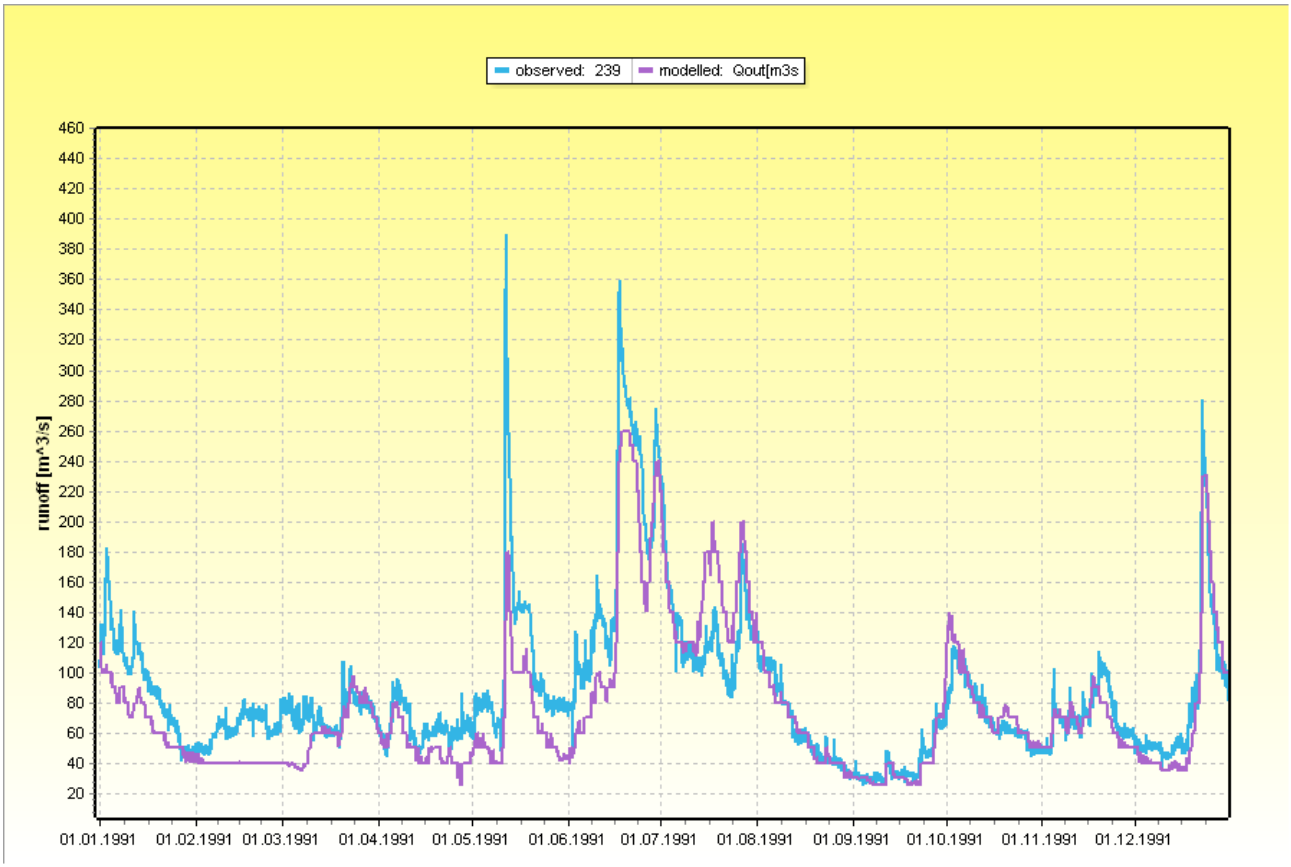


Abbildung 2.19.1: Comparison of modeled (purple) and observed runoff (blue) for lake Zurich without externally read abstraction values (purely modeled)

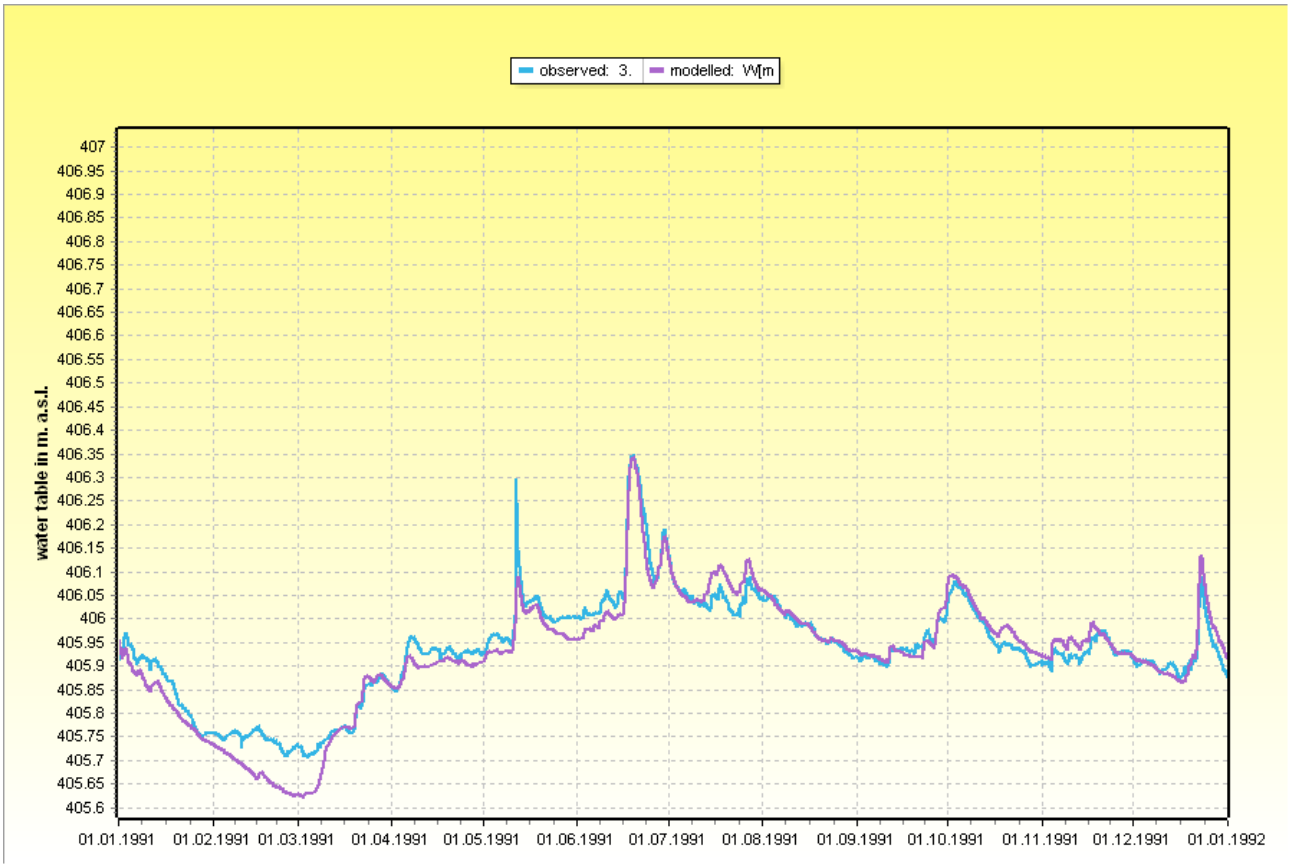


Abbildung 2.19.2: Comparison of modeled (purple) and observed water levels (blue) for lake Zurich without externally read abstraction values (purely modeled)

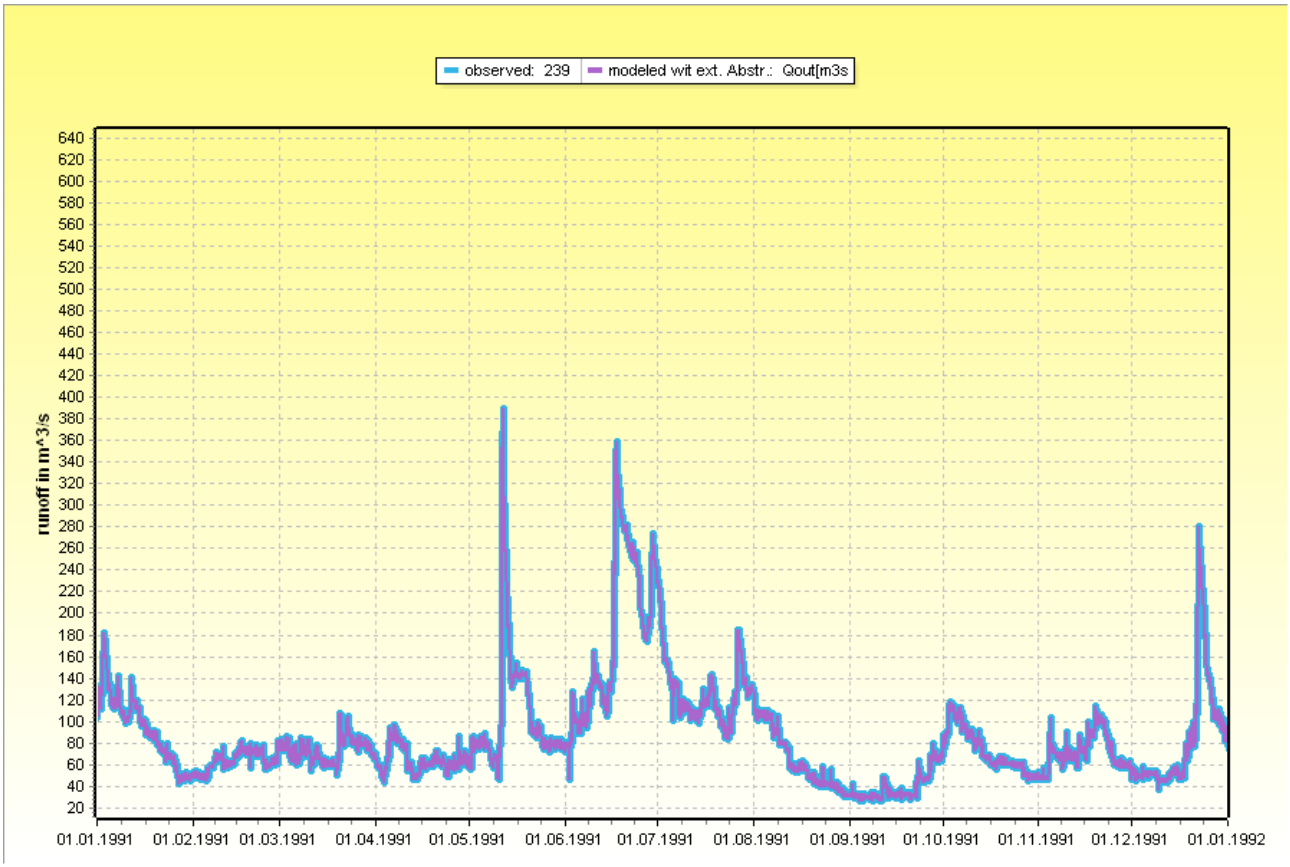


*Abbildung 2.19.3: Comparison of modeled (purple) and observed runoff (blue) for lake Zurich **with** externally read abstraction values (only external values used)*

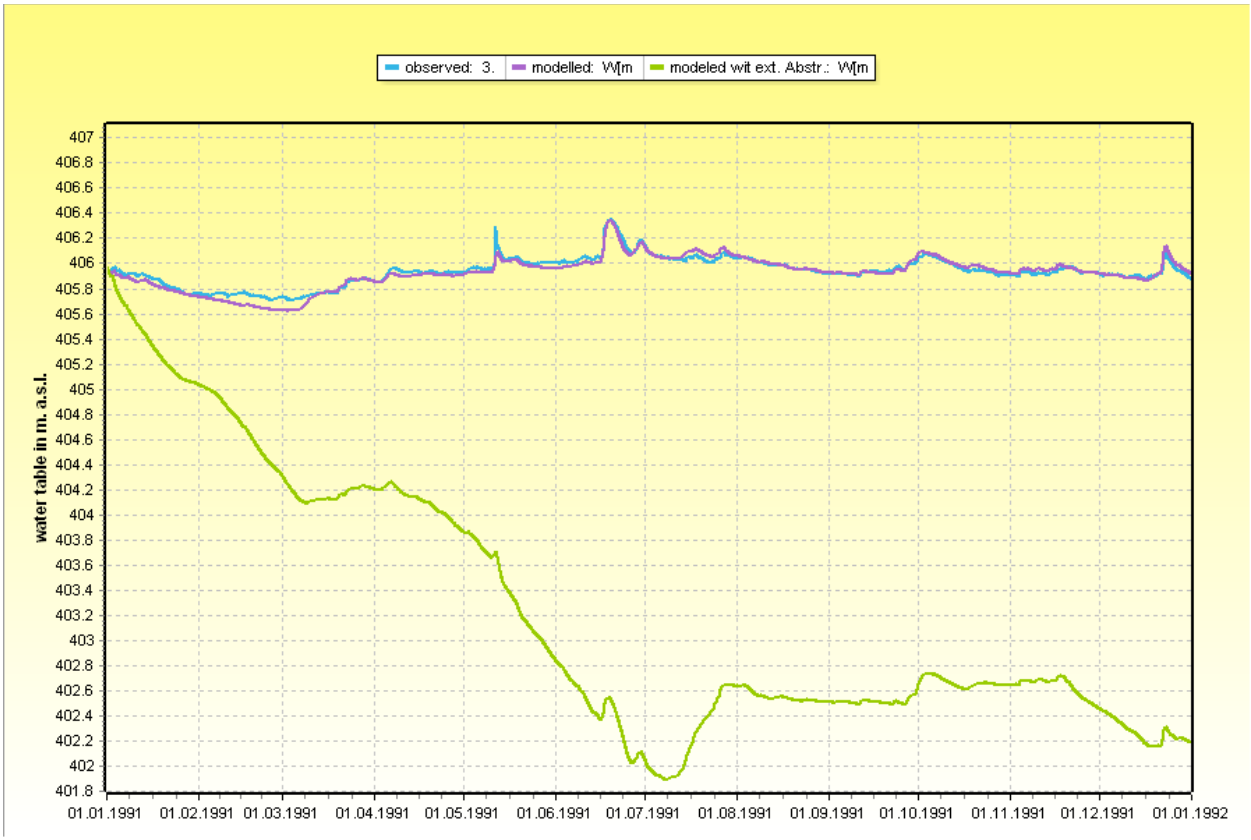


Abbildung 2.19.4: Comparison of modeled and observed water tables for lake Zurich without (purple) and with externally read abstraction values (green)

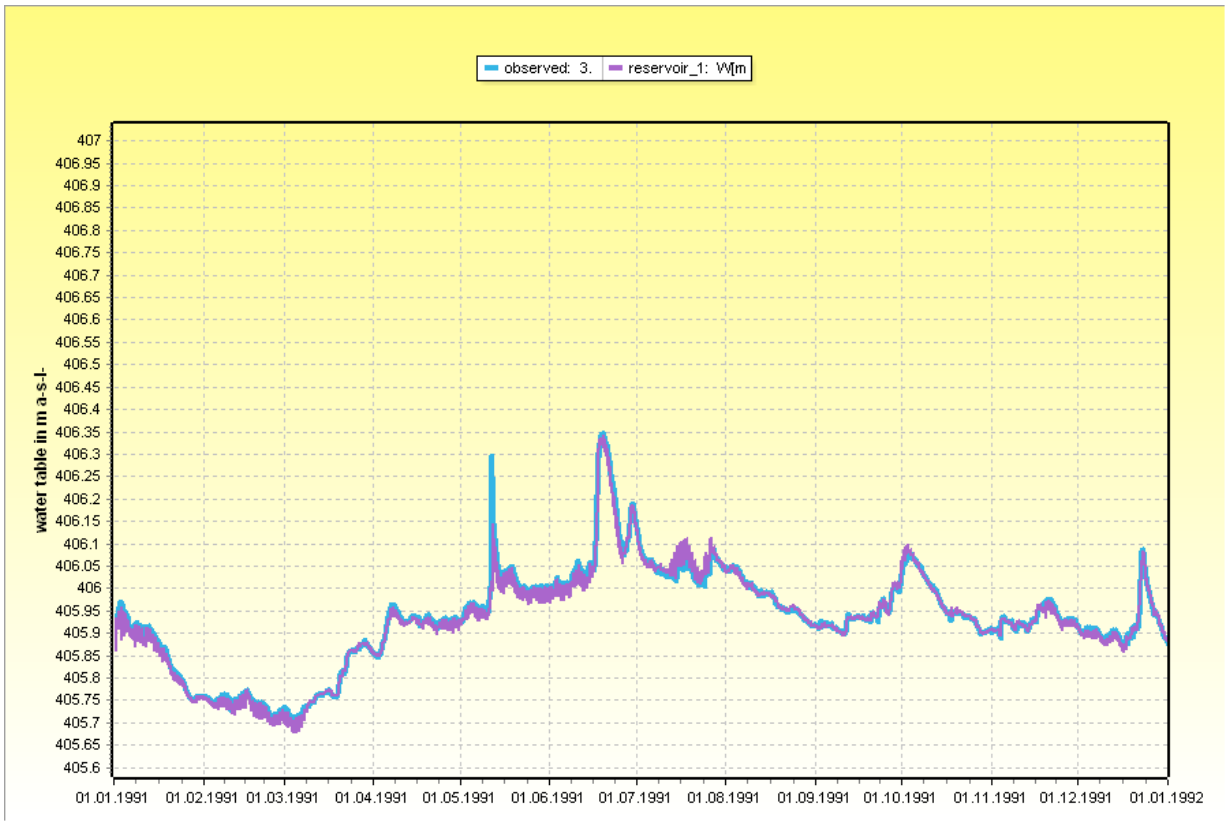


*Abbildung 2.19.5: Comparison of modeled (purple) and observed water levels (blue) for lake Zurich **with** externally read abstraction values and **with** externally assimilated water table values (only once a day)*

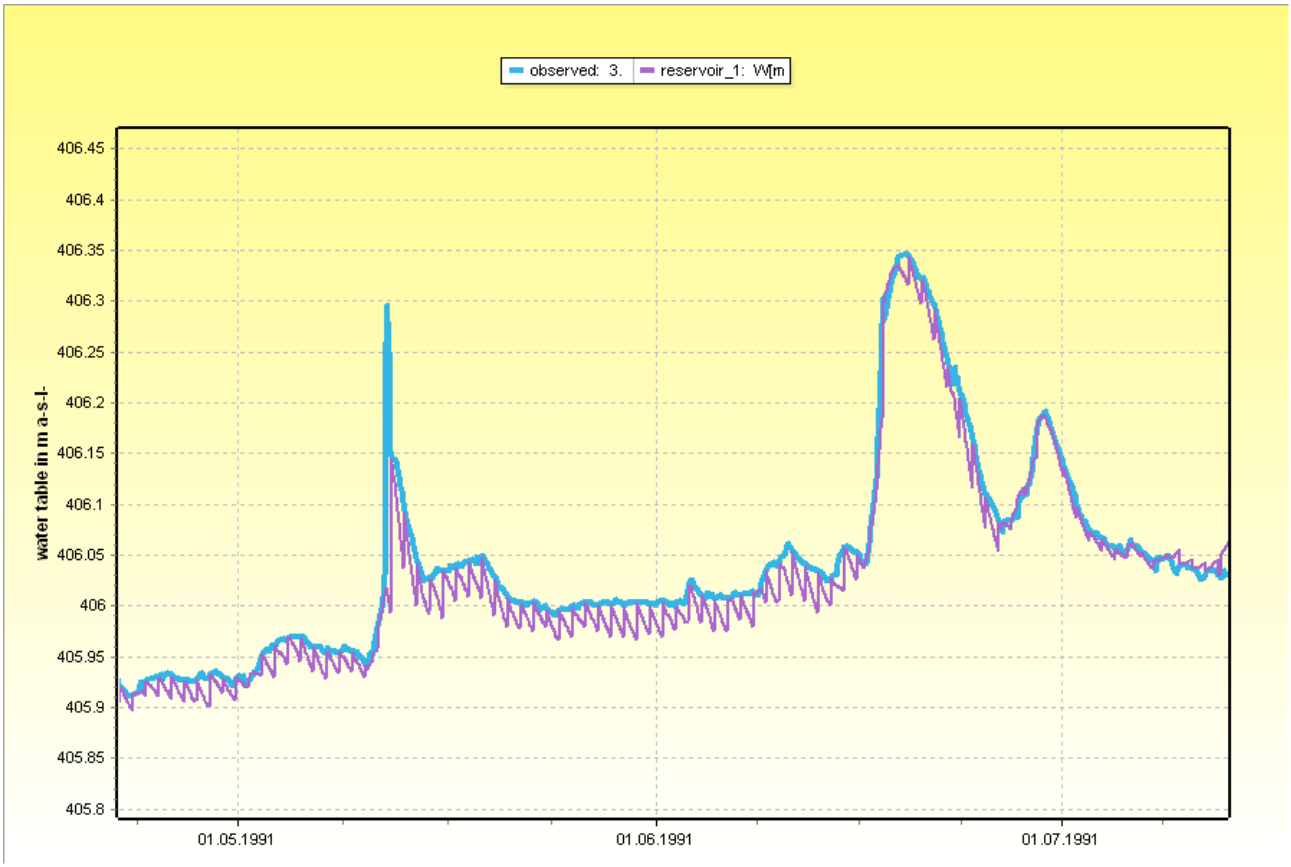


Abbildung 2.19.6: Comparison of modeled (purple) and observed water levels (blue) for lake Zurich with externally read abstraction values and with externally assimilated water table values (only once a day), zoom-in

2.20 Performance criteria R^2 and EV

In order to evaluate the performance of the model a series of performance criteria is calculated. For this it is required to incorporate observed runoffs into the modeling process. The outlets of selected modeled subcatchments should match the locations of the flow gauging stations at the rivers. The first performance criteria is the R^2.

$$R^2 = 1 - \frac{\sum_i \epsilon_i^2}{\sum_i \left(x_i - \bar{x}\right)^2} = 1 - \frac{\sum_i \left(y_i - x_i\right)^2}{\sum_i x_i^2 - \frac{1}{n}\left(\sum_i x_i\right)^2} \qquad \textbf{(2.19.1)}$$

with y_i simulated value (e.g. runoff [mm])
x_i, $\bar{x}$ observed value (e.g. runoff [mm]), mean observed value
ε_i derivation observed - modeled (y_i - x_i) at time step i
n number of time steps used for calculating R^2

In order to be able to recognize impacts of a systematic time shift of the modeled to the observed data the performance criteria are calculated as a series of 21 values shifted by ± 10 time steps with respect to the central value. The location of the maximum R^2 within this series can be useful to re-calibrate routing parameters.

Because runoff data usually does not fulfill the theoretical requirement of normal distribution the impact of flood peaks is over estimated. If the purpose of the modeling is in simulating low flows or

the entire flow spectrum the performance criteria should be taken from normal distributed data sets. Therefore in WaSiM there is the possibility to calculate a performance criteria of the logarithms of the runoff values.

Another performance criteria is the explained variance (*EV*). It is able to eliminate the impact of systematic shifts of the data along the y-axis in the performance criteria (e.g. systematically to large or to small modeled data, even if they are right in time). A comparison between the *EV* and the R^2, which itself does not consider systematic shifts, may show these effects. If the explained variance *EV* is larger than the corresponding R^2, then it is highly probable that there are systematic errors in the modeled data.

$$EV = 1 - \frac{\sum_i \left(\epsilon_i - \mu_e\right)^2}{\sum_i \left(x_i - \bar{x}\right)^2} = 1 - \frac{\sum_i \epsilon_i^2 - \frac{1}{n}\left(\sum_i \epsilon_i\right)^2}{\sum_i x_i^2 - \frac{1}{n}\left(\sum_i x_i\right)^2} \qquad \textbf{(2.19.2)}$$

with ε_i derivation of the modeled from the observed value (y_i - x_i) like in equ. (2.19.1)

μ_ε mean derivation of the modelled from the observed values = E(ε_i)

x_i, $\bar{x}$ observed value at time step *i* (e.g. runoff [mm]), mean observed value

n number of time steps used for *EV*-calculation

Like for the R^2 it is also possible for the EV to use the logarithms of the runoff values. Thus there are in fact 4 types of performance criteria: the R^2 for linear and for logarithmic values and the explained variances EV for linear and logarithmic values. Each of these criteria is calculated for a time shift of ± 10 time steps resulting in a total number of 4 x 21 = 84 criteria numbers for each subbasin.

2.21 Coupling the solute transport to the water fluxes

In WaSiM version 2 the modeling of ideal tracers in included. It is possible to consider radioactive decay, mixing and enrichment (for salts). Only advective transport and mixing is considered, no diffusion is taken into account. This is a useful simplification if diffuse sources like salt from irrigation water or tritium from precipitation is modeled. Salts are modeled using the absolute masses, all other tracers are modeled by their concentrations. The advantage is, that for salts no changes in the masses will occur during evaporation while for evaporating tracers also no changes in their concentrations will occur. The mixing of water volumes with different concentrations is calculated after:

$$c_{mix} = \frac{c_1 V_1 + c_2 V_2}{V_1 + V_2} \qquad \textbf{(2.20.1)}$$

with V_1, V_2 volumes 1 and 2 to be mixed

c_1, c_2 concentrations of the tracer in the volumes 1 and 2

c_{mix} mixing concentration

Radioactive decay is calculated using the half time $t_{½}$, which is converted into a decay constant depending on the model time step. Thus the radioactive decay is computed by:

$$c_i = c_{i-1} \cdot e^{-\Delta t / t_{1/2} \cdot \ln 2} \qquad \textbf{(2.20.2)}$$

with c_i concentration of the radioactive tracer after decay

c_{i-1} concentration of the radioactive tracer before decay (from the last time step)

Δt time step [days]

$t_{1/2}$ half time [days]

There can be considered at maximum 9 tracers at the same time. The tracers are modeled absolutely independent of each other even if the concentrations are very high. This is a limitation as well as the inconsideration of the fact that the suctions and conductivities of salty solutions differ from that of pure water. Also the effects of density on the fluxes are not considered. So the model should not be applied to systems where such effects are expected to be important for the water fluxes.

Note: Solutes transport and concentrations cannot be calculated for multi-layered vegetations in the interception model.

2.22 Glacier model

WaSiM offers two methods for modelling glaciers. Here are some commons and differences:

- The traditional static glacier model was implemented in 1999
  - this is mainly a model describing glacier melt and discharge accumulation and routing within the glacier in a conceptual way.
  - No changes in glacier extend, no explicit mass balance (i.e. total ice mass calculation) and no metamorphosis of snow to firn and ice is modelled. Thus, the extend of a glacier can and must be statically defined.
  - Each glacier cell is defined by a code in a glacier grid (with separate codes for ice and firn).
  - For each separate glacier, a separate sub-basin must be defined in the sub-basin grid – accordingly, the routing model must be extended by routing descriptions for those glacier sub basins. A glacier subbasin mustn't contain any unglacierized cell
- The dynamic glacier model was implemented in 2010
  - it uses exactly the same melt methods as the static glacier model (t-index with or without radiation correction)
    - the latest versions now offer the use of other snow melt routines (like the ones for unglacierized cells), so the melt rates are comparable, especially for partly glacierized cells
  - however, the glacier extend is dynamic: depending on the annual balance (at a given day, e.g. September, 30th) the extend of the glaciers will shrink or grow based on a relatively simple relation, the A-V-relation, see also Stahl et al. (2008)
  - An explicit balance grid is calculated, allowing the shrinking and growing calculations as well as other estimations associated with glacier mass changes.
  - Since the glaciers can vary in their extend, the fixed relation of one sub basins per glacier is not applicable here. Instead, each sub-basin may contain any number of glaciers (which are identified as in the static model by their code).
  - Each single ice body or glacier can be defined in the glacier code grid with a unique code, so the A-V-relation is applied to each separate glacier, allowing for different behaviors of different glaciers depending on their altitude, precipitation and temperature.
  - In addition to the melt model, the dynamic glacier model allows for modelling the metamorphosis of snow to firn (which is done once per model year, when resizing the glacier), so the glaciers mass can change not only by melt but also by precipitation – this is quite important for the algorithm to work, since an explicit balance combined with modelled glacier melt would mean that the ice mass will be smaller year after year if there weren't sources of new ice as well.

The dynamic glacier model can be seen as an advancement to the static glacier model. By allowing the glacier size to be changed, it provides a possibility to calibrate the glacier model not only at discharges but also at observed historic glacier extends. Also, the future projection of glacier extends is one of the central practical applications of such an approach.

2.22.1 Static glaciers

The static glacier option assumes that the glacier hypsometry and extent remains constant although in reality the glacier may retreat or advance. The melt model describes the melt of firn, ice and snow on glaciers as well as the routing of the glacier discharge within the glacier. The module is fully integrated in the snow model and it can be used with both model versions (Topmodel and RICHARDS-approach).

To enable the glacier module, an additional standard grid, the so called glacier grid, has to be read in. It codes the cells which are covered by glaciers with the codes 1 for ice and 2 for firn, respectively. Further it is required that glaciers are defined in the subbasin grid as separate subcatchments. Thereby, it is not necessary that each small glacier must be a separate subbasin, it is perfectly possible to put some small glaciers together into one subbasin (glacier subbasins do not have to be closed areas). However, it is required that the glacier subbasins as defined in the subbasin grid does not contain any other landuse than firn or ice and that both, firn and ice, are present only in glacier subbasins, thus the glacier subbasins are exclusive for the glacier melt model. This must hold because of the internal model branching since all of the following submodels are skipped for glacierized cells when using the static glacier model (note: this is not always the case when using the dynamic approach):

- interception
- soil model (RICHARDS-approach or TOPMODEL-approach)
- irrigation control
- real evaporation
- groundwater flow
- solute and tracer transport in all of the above mentioned modules

Only the discharge routing re-unites the data streams again, because discharges of glacier subbasins are routed like any other discharges. A section "[ice_firn]" was inserted in the control file containing all melt and runoff concentration parameters and file names for the glacier model. In addition to the statistics output file and the grid for the snow melt, there will be also grids and statistics files for melt of ice and firn. A fourth statistic file contains the total glacier melt. This last statistics file contains the already internally routed and superposed discharges, whereas the snow, firn and ice melt statistics files contain the areal averaged melt rates for each time steps without reservoir effects. All of these statistics files contain data columns for each subbasins, even if there are no glaciers, in which last case the data are only zero for firn and ice melt.

melt of firn and ice:

There are two methods for calculating glacier melt: a classical temperature index method and an extended temperature index method after Hock (1998, 1999) including information on the global radiation during each time step on each grid cell in order to modify the melt. The melt after the classical temperature index approach is calculated by:

$$M = \begin{cases} \frac{1}{n} DDF_{snow/ice/firn} \cdot (T - T_0) & T > T_0 \\ 0 & T \le T_0 \end{cases} \qquad \textbf{(2.21.1)}$$

with M melt [mm/time step]
n number of time steps per day [day^{-1}]
DDF 3 degree-day-factors for snow, firn and ice, respectively [$mm \cdot °C^{-1} \cdot day^{-1}$]

T air temperature in a standard elevation of 2 m [°C]
T_0 threshold temperature for melt [°C]

The melt is calculated for snow, firn, and ice separately using specific parameters. If a glacier cell, which is recognized by the codes 1 or 2 in the ice_firn-grid, is covered with snow, the melt calculation is done using the parameters DDF_{snow} and no firn or ice melt is calculated. If the cell is not covered by snow, then the melt calculation is done using the parameters according to the cell code: DDF_{ice} for ice cells (code 1) or DDF_{firn} for firn cells (code 2). Usually the relation $DDF_{ice} > DDF_{firn} > DDF_{snow}$ holds. All three parameters are constant in space and time for the entire model domain. The parameter values are ranging between 2 … 10 mm·°C^{-1} ·day^{-1}.

Using the extended melt approach after HOCK (1998), the melt of snow, firn, and ice is calculated by:

$$M = \begin{cases} \left(\frac{1}{n} MF + \alpha_{snow/ice/firn} \cdot I_0 \cdot \frac{G_s}{I_s} \right) \cdot (T - T_0) & T > T_0 \\ 0 & T \le T_0 \end{cases} \qquad \textbf{(2.21.2)}$$

with M melt [mm/time step]
n number of time steps per day [day^{-1}]
MF melt factor with identical values for snow, firn and ice (around 2 mm·°C^{-1} ·day^{-1})
α empirical coefficients for snow and firn (identical, app. 0.002) and for ice (around 0.001) [mm·Wh^{-1} ·m^2 ·°C^{-1} ·day^{-1}]
I_0 potential direct incoming shortwave radiation (visible incoming sun radiation) for each grid cell (depends on shading, aspect and slope angle, time and season, but not on cloudiness) [Wh·m^{-2}]
I_S like I_0, but for the well defined location of a meteorological station [Wh·m^{-2}]
G_S observed radiation at the same station [Wh·m^{-2}]
T air temperature in a standard elevation of 2 m [°C]
T_0 threshold temperature for melt

The coefficient is multiplied by global radiation, hence increasing melt when shortwave radiation input is high and vice versa. For each grid cell the interpolated global radiation is used, computed using equation (2.5.11) . The actual global radiation correction is done in the evaporation module. However, for this melt model approach, it is necessary to enable the radiation correction and to use PENMEN-MONTEITH or WENDLING evaporation in order to get corrected radiation. Also it is recommended not to use net radiation instead of global radiation. If no radiation correction and shading is done, no variations in the melt pattern will be recognizable since only the interpolated radiation is used for scaling the melt. The model stops it's execution if no radiation is found during the initialization.

The parameters α_{ice} , α_{firn}, and α_{snow} are becoming smaller in this order and range between 10^{-4} … $5 \cdot 10^{-3}$ mm·Wh^{-1} ·m^2 ·°C^{-1} ·day^{-1}. The radiation is given in Wh·m^{-2} scaling the small α-values to the order of reasonable degree day factors. Since the product of α and interpolated global radiation is added to the melt factor *MF*, *MF* is smaller than the degree day factors for a classical temperature index approach (Equation (2.21.1)) and it's value is around 2 mm·°C^{-1} ·day^{-1}.

If a glaciated cell is coded as firn but covered with snow, the melt was calculated using the parameters for snow. However, since this snow melt flows directly into the firn layer and since the reservoir effects of firn are so much larger than the reservoir effects of snow, the melt of snow on

firn cells is handled like melt of firn.

The radiation coefficient vary over time. The control file must provide the minimum and the maximum value for the radiation correction factors ($\alpha_{ice,min}$, $\alpha_{ice,max}$ and $\alpha_{snow,min}$, $\alpha_{snow,max}$), which are then internally used to compute the effective radiation correction factors for any given Julian day:

$$\alpha = a + b \cdot \sin\left(c \cdot \left(d + JD\right)\right) \qquad \textbf{(2.21.3)}$$

with α the respective correction factor for snow, ice or firn
a mean value $(\alpha_{min} + \alpha_{max})/2$ of the respective α-parameters
b amplitude $(\alpha_{max} - \alpha_{min})/2$ of the respective α-parameters
c conversion of the Julian Day number into radiant measure: $c = 2\pi/365 = 0.017214$
d shift of the *sin* () function to give the result of 0 at March, 21 st and September 21 st and
1 at June, 21 st and -1 at December 21 st

routing of the different melt components through the glacier:

For glacierized subbasins there are three runoff components calculated after equations (2.21.1) and (2.21.2) , respectively. These components are now superposed and routed to the subbasin outlet using a set of parallel single linear reservoirs.

The total runoff from a glacier subbasin is given by:

$$Q(t) = \sum_{i=1}^{3} \left(Q_i(t-1) \cdot e^{-\frac{1}{k_i}} + R_i(t) \cdot \left(1 - e^{-\frac{1}{k_i}}\right) \right) \qquad \textbf{(2.21.4)}$$

with $Q(t)$ total runoff during time step t [mm/time step]
i reservoir index (1 reservoir for snow melt, 1 for firn melt and 1 for ice melt)
$Q_i(t\text{-}1)$ outflow of reservoir i during the last time step t-1 [mm/time step]
$R_i(t)$ input into reservoir i in the actual time step t, this equals the sum of melt and additional rain [mm/time step]
k_i storage coefficient (recession constant) for reservoir I

Equation (2.21.4) describes the superposition of three parallel reservoirs 1...i (i=3), each of which has specific storage coefficients k_i and inflow rates R_i, representing the areal means of the three melt components. The storage coefficients for ice, snow and firn range in this order with values for k_{firn} of around 100 to 1000, for k_{snow} of 10 to 100, and for k_{ice} of 1 to 20 hours.

For the calibration of the model it is important to initialize the three parallel reservoirs with matching initial values. If the calibration starts at a state when the total glacier is frozen, e.g. in January, then only the firn reservoir must be initialized with a small initial outflow, the other two reservoirs may be initialized with zero.

The total runoff from the glacier subbasins is handled in the discharge routing module like the runoff of any other subbasin. Thus, WaSiM can be applied also to very large basins ranging from glacierized high mountains down to low land regions with important groundwater reservoirs, lakes and irrigation impacts.

The control file must contain a section named [ice_firn], if an ice_firn-grid is given as a standard grid:

```
[standard_grids]
<n>      # number of standard grids
# path     # identification # fillcode 0=no, 1=yes (fill with values of nearest neighbor)
```

```
$inpath//$ice_firn_grid         ice_firn      0 # grid with firn or ice cells
…
[ice_firn]
2 # method for glacier melt: 1=static with t-index, 2= static with t-index and radiation
# correction, 11 = dynamic glaciers with t-index, 12= dynamic glaciers with t-index and
#radiation correction
5                    # t-index factor for ice
4              # t-index factor for firn
3              # t-index factor for snow
2             # melt factor
-0.00010      # radiation coefficient for ice_min  (for method 2/12)
+0.00070      # radiation coefficient for ice_max  (for method 2/12)
+0.00006      # radiation coefficient for snow_min (for method 2/12)
+0.00030      # radiation coefficient for snow_max (for method 2/12)
12            # single linear reservoir (SLR)-constant for ice
120           # single linear reservoir (SLR)-constant for firn
24            # single linear reservoir (SLR)-constant for snow
1             # initial reservoir content for ice discharge (SLR approach)
1             # initial reservoir content for firn discharge (SLR approach)
1             # initial reservoir content for snow discharge (SLR approach)
$outpath//$firn_melt # melt from firn
$Writegrid             # 0, if no grid-output is needed, else one of the codes described
above
$outpath//qfir//$grid//.//$code//$year $hour_mean # melt from firn as statistic file
$outpath//$ice_melt  # melt from ice
$Writegrid     # 0, if no grid-output is needed, else one of the codes described above
$outpath//qice//$grid//.//$code//$year $hour_mean # melt from ice (statistic)
$outpath//qglc//$grid//.//$code//$year $hour_mean # melt from ice and firn (statistics)
…
```

2.22.2 dynamic glaciers

The dynamic glacier model does not only model the glacier mass balance but also advance or retreat of the glacier by removing or adding glacierized grid cells or fractions thereof. The model does not account for changes in surface elevation of glacier grid cells, only the extent of the glacier.

As for the static approach , first the glacier mass balance has to be calculated . This is done by balancing mass gain (snow accumulation) and mass losses (snow, firn, ice melt) over one mass-balance year (usually October 1st to September 30th). The month and day for calculating the balance (and the glacier advance and retreat) are parameters to the dynamic glacier model as well. It is recommended to chose the parameters to date when most old snow is molten and the new snow season didn't start already, so e.g. September 30th may be a good choice under northern hemisphere conditions.

Equation (2.21.5) will be used when initializing WaSiM with a grid containing glacierized cells under cold start conditions. This is one of two new input grids. The glacierized cells may have any value between 0 and 1, so a cell may also be fractionally glacierized. Beside modelling the firn layers, this is the other important improvement compared to the method presented in Stahl et al. (2008). With this range of glacier coverage between 0...1 it is possible to use the dynamic glacier model with smaller resolutions (e.g. 100m to 500m). Also, an annual balance is possible (and not only one balance per decade).

Modelling glacier retreat and advance

Glacier area changes are modeled using volume-area scaling

$$V = b \cdot A^{f} \qquad \textbf{(2.21.5)}$$

with V volume of the glacier in m·km^2
A area of the glacier in km^2
b empiric factor
f scaling factor assumed to be 1.36

This relationship (2.21.5) was first presented by Chen and Ohmura (1990) and physically validated by Bahr et al. (1997). The parameters b and f must be provided in the parameter file. A value of b=28.5 m is suggested (would be the mean ice thickness of a 1 km ^{2} glacier) but other values may be chosen if data are available. .

By using equation (2.21.5) , the newly calculated WE value of the mass balance - converted into a volume by multiplying the resulting mass with the ice density of approx. 918 Kg/m^3 - can be used to calculate the new area for each glacier by equation (2.21.7) :

$$A = \left(\frac{V}{b}\right)^{\frac{1}{1.36}} = \left(\frac{V}{b}\right)^{0.735} \qquad \textbf{(2.21.7)}$$

The “tricky” thing is to add or remove fractions of grid cells to or from the glacierized area. Glacierized areas usually grow/shrink at the tongue faster than in the accumulation area, retreat should not occur in areas with a positive balance, so the area covered by firn should also not be affected by the retreat at the edge of the glacier as long as there are cells which are completely free of firn. Also, the mean temperature should be regarded when changing the glacierized areas. To take all this into account, the area change is first converted into the number of grid cells to be added or subtracted from the glacier. The next step is to divide the glacier into elevation belts and to reduce or increase the number of gridcells (or the ice covered fraction) according to the elevation. This process is done iteratively. The glacier is partitioned into elevation bands of equal elevation differences (how many belts there will depends on a control file parameter – upper limit is the equilibrium line if there are cells without firn, else the entire glacier is used for growing/shrinking).

Only the lowermost elevation band will be processes in the first iteration step. The next elevation band will be part of the iteration starting with the second step, the third band will be handled for the first time in the third iterations step and so on, such that the uppermost elevation band will be affected by growing or shrinking in the last iteration step only whereas the lowermost elevation band is affected in every iteration step. In addition, for the lowermost elevation belt an additional number of iterations (extraWeightFactorBand0) can be set in order to pronounce the advance/retreat at the glaciers tongue.

Example: A glacier has an elevation range from 1500 to 2500m a.s.l. The equilibrium line lies at 2100 m a.s.l. The control defines: Number of iterations: 10; extraWeightFactorBand0=4. The model runs with a resolution of 500 m.

The elevation range between 1500 and 2100 m is divided into 10 elevation bands of 60m each. When the new area is calculated, there should be 10 cells be subtracted from the rim of the glacier. The count of cells at the glaciers edge are (as an example):

band 1: 8 cells, affected in each of the 10 iteration plus 4 extra iterations: 8 x (10x4) = 320
band 2: 13 cells, affected in iterations 2 to 10 = 13 x 9 = 117
band 2: 15 cells, affected in iteration 3 to 10 = 15 x 8 = 120
band 2: 14 cells, affected in iteration 4 to 10 = 7 x 14 = 98
band 2: 14 cells, affected in iteration 5 to 10 = 6 x 14 = 84
band 2: 12 cells, affected in iteration 6 to 10 = 5 x 12 = 60
band 2: 18 cells, affected in iteration 7 to 10 = 4 x 18 = 72
band 2: 7 cells, affected in iteration 8 to 10 = 3 x 7 = 21
band 2: 10 cells, affected in iteration 9 to 10 = 2 x 10 = 20
band 2: 9 cells, affected in iteration 10 only = 9

in total: 120 cells are affected, whereas the 10 iterations will split the 10 cells into 921 fractions. So each affected cells glacierized fraction has to be decreased by 10/921xnumber of iteration. A single

change amount is 10/921 = 0.01086. From each edge-cell in band 1 the amount of 40x0.01086 = 0.43 is subtracted, so 3.47 cells are removed from the glacier in band 1 already. From band 2, there have to be removed the area of 117x0.01086 cells = 1.27 cells, spread over all 13 edge-cells and distributed over iterations 2 to 10 and so on.

This schema could have been run without iteration at all. But: since the exact geometry is not known to the algorithm, the shrinking and growing could lead to a changing number of cells during the process of retreat or advance. Therefor the algorithm runs iteratively, changing only a small amount of area in each iteration. If the number of edge cells changes, then the total of removed fractions will be smaller than computed first. So the amount to be removed from each cell (or to be added to each cell) can be changed dynamically from iteration step to iteration step.

The entire procedure is quite fast, since it is done only once a year and for a rather limited number of glaciers (even if there were hundreds of glaciers). So there is no sensible negative performance impact of the dynamic glacier model on the total model performance (in terms of computing speed).

Firn model

The dynamic model also includes a dynamic firn model, i.e. changes in firn depth are calculated at the end of each mass-balance year using a simple approach of m*etamorphosis of snow to firn to ice:*

Snow which does not melt within one balancing period will automatically become firn at the beginning of the next balancing period. When this firn layer is covered by snow in the following winter and only partially melts in the following summer, then another firn layer forms. During the following year, another layer of snow covers the now two (numeric) layers of firn. This process will eventually lead to the firn turning into ice. Usually, this metamorphosis will take several years The number of years needs to be defined in the model (default is seven, i.e. after seven years old snow becomes finally ice). These grids are stacked together to a so-called firn stack, which can (and must) be initialized in cold start conditions by providing some parameters:

- Number of years it takes the snow to convert into ice (recommended default: 7)
- Elevation of the equilibrium line in m a.s.l. This is the elevation where the amount of precipitation (as snow) equals the amount of snow melt, so the mass balance is zero there. All elevations above have a positive net balance (more snow fall than melt), all elevations below have a negative balance (more melt than snow fall). Typical elevations for the equilibrium line in the Swiss Alps are between 2300 and 2800 m a.s.l.
- To be able to initialize the amount of firn for each stack layer, the last parameter defines the rate of change in the water equivalent (WE) of firn with each meter of elevation rise. This is typically something between 1 and 2 mm/m, so the WE value of firn rises by 100 to 200 mm per 100 m elevation rise.

After each mass-balance year the firn stack will be adjusted in thickness according to the mass balance, i.e. the total firn depth (in water equivalent) may change. Changes may occur in the top layer(s) and at the bottom of the firn pack. In case of a positive mass balance (i.e. not all winter snow has melted), a new firn layer is added equal in thickness (as water equivalent) to the mass balance. In case of a negative mass balance, firn layers are removed or decreased in thickness according to the excess melt once the snow is molten. Depending on the mass balance several of the uppermost firn layers may completely melt as well. In extreme situations it may even occur that all firn layers will completely melt. If a firn layer melts away the layer is technically not removed but given a thickness of 0 m water equivalent.

In addition, the lowermost layer is removed from the firn pack assuming it to become ice. Consequently, since the uppermost layer is newly created each year, the number of firn layers remains the same and after each year firn layers move one layer down, leaving the uppermost layer to be replaced by the WE amount of the snow which did not melt during the current mass-balance year.

Since glacierized cells doesn't need to be glacierized completely (the cell values of the grid provided with the glacierized fraction may vary between 0 and 1), a problem occurs when snow is converted to firn on the glacierized fraction: This mustn't be done on the unglacierized fraction of the cell. Consequently, the snow model must keep track of both compartments in parallel. That lead to the introduction of some new grids (starting from version 9.07.00), holding information on the snow storage (liquid, solid, total), the snow pack temperature, the surface temperature, the snow melt rates and the snow evaporation. Since all these grids are needed for the dynamic glacier model only, they are created or read in when using the glacier model only.

Control file changes for the dynamic glacier model

There are a few additional parameters as well as name definitions for additional statistic outputs and grids:

```
[variable_grids]
2      # Number of variable grids to read
$outpath//glca.grd GlacierizedCells 0  -9999     # glacierized fraction
7 # $Writegrid
1 # $readgrids
$outpath//$glid    GlacierCodes     0  -9999     # unique codes for each glacier
7 # Writegrid
1 # $readgrids

…

[ice_firn]
... #here the entries of the static glacier model must follow
# ------------------------------------------------------------------------
# now some new parameters for the new dynamic glacier model (methods 11 and 12)
$outpath//qsgl.stat $once_per_interval     # melt from snow from glacier only
$readgrids # 1=read grids and stacks from disk, 0=generate new grids and stacks
#(using the parameters in the following line for WE_Firn stack)
7 2700 1.8 # number of layers for the firn stack, followed by two
# initialization parameters: average Equilibrium line elevation in m
# (e.g. 2500) and change rate of WE per m in mm (e.g. 2) -> every 100m
# the WE of firn in each layer will grow by 200mm
09 30 # month and day (hour is set automatically to 24) for which the Volume-
# Area-Relation will be applied newly (and temporary (i.e. internal) Balances
# are reset to 0)
28.5 1.36 10 4 # VAscaling and VAexponent for Volume-Area-Relation of glaciers
# and number of iterations (elevation belts) and
# extraWeightFactorBand0 (elevation band 0 will be processed in each
# iteration this given number of times more than once. Default = 0)
$outpath//$firn_WE_stack # water equivalent for firn (given as stack, number of
# layers taken from the parameter given before); layer 0 will contain the total
# WE for all firn layers
$Writestack # write code for the Firn stack
$outpath//glfirn.stat $once_per_interval  # water equivalent for firn as
# statistics file (sum over all firn layers)
$outpath//glmb.grd     # output grid with mass balance of the glacier
$Writegrid   # writecode for the glacier mass balance
$outpath//glmbo.grd      # output grid with mass balance of the glacier (old
year)
$Writegrid   # write code for the glacier mass balance (old)
$outpath//glmb.stat $once_per_interval # mass balance for the glaciers as
# statistics file (mass balance for each time step with respect to the
# entire subbasin the glaciers are located in)
$outpath//glmb2.stat $once_per_interval    # mass balance for the glaciers as
# statistics file (mass balance for each time step with respect to the glaciers
# only!)
1.0  #  additional  parameter  when  using  a  debris  grid:  this  value  is  used
#  to  globally  scale  the  values  of  the  debris  grid.  Only  values  >  nodata
```

```
# are regarded, i.e. when a cells melt coefficient should  not be altered, the
# debris grid should contain -9999 at this location
# ----------------------------------
# some new parameters for better statistic output (for balance verfication) and
# also separate handling of snow on glaciers and beside glaciers on the same
# cell
$outpath//ssto_OffGlac_//$grid//.//$code//$year $once_per_interval # total snow
# storage, in mm, (liquid and solid fraction) for the unglacierized part of
# a cell (usefull for annual balances of all inputs/outputs/storages, since
# snow on glaciers is handled in the glacier mass balance already)
$outpath//$SSNOOnGlacgrid    # name of the grids with the snow storage solid in
# mm valid for the glacierized part of a cell
1//$Writegrid # one of the grid write codes, at least last grid is useful
$outpath//$SLIQOnGlacgrid     # name of the grids with the snow storage liquid
in
# mm valid for the glacierized part of a cell
1//$Writegrid # one of the grid write codes, at least last grid is useful
$outpath//$SSTOOnGlacgrid  # name of the grids with the total snow storage
# solid AND liquid in mm valid for the glacierized part of a cell
1//$Writegrid # one of the grid write codes, at least last grid is useful
$outpath//$QSNOWOnGlacgrid # name of the grids with the total snow outflow in
# mm valid for the glacierized part of a cell
1//$Writegrid # one of the grid write codes, at least last grid is useful
$outpath//$snowsurftemperaturgridGL       # name of the grids with snow surface
# temperature, valid for the glacierized part of a cell
1//$Writegrid # one of the grid write codes, at least last grid is useful
$outpath//$snowtemperaturgridGL        # name of the grids with snow pack
# temperature, valid for the glacierized part of a cell
1//$Writegrid # one of the grid write codes, at least last grid is useful
$outpath//$ETRSgridGl                # name of the grids with snow evaporation
in
# mm, valid for the glacierized part of a cell
1//$Writegrid # one of the grid write codes, at least last grid is useful
```

example for dynamic glacier model results

Figures 2.22.2.1 and 2.22.2.2 show the model results for the Swiss Alps around the Aletsch-Gletscher (the one with the long tongue in the center) between 1996 and 2010. Brown cells indicate areas where the glaciers have retreated according to the parameters used in the model setup and the meteorologic input used. For this example, some extreme parameters have been used in order to show the effect during such a short modeled time of 15 years.

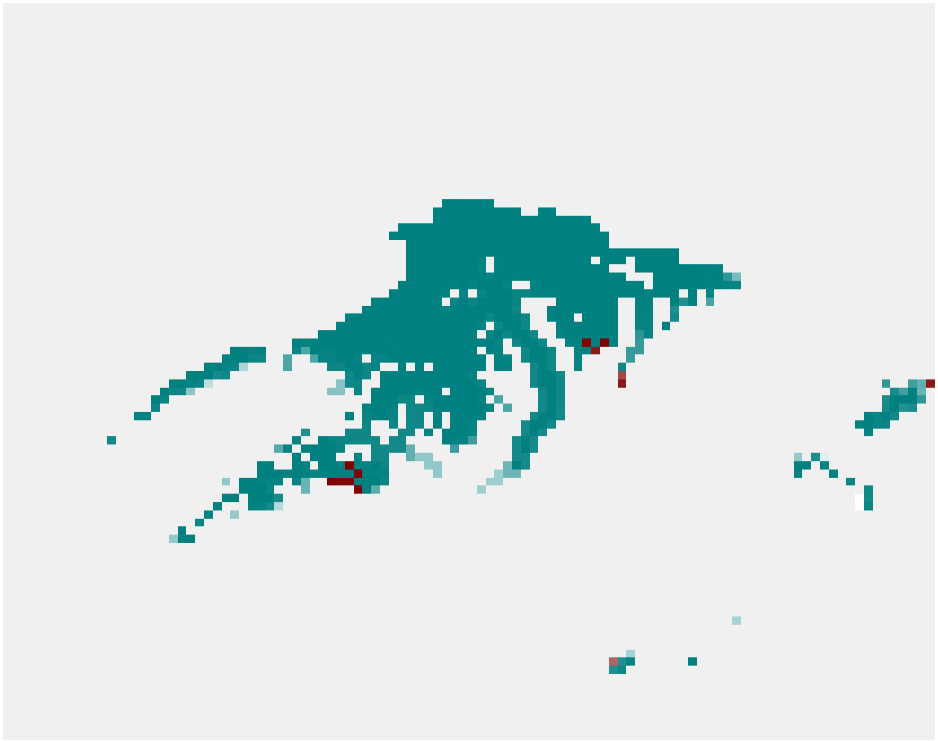


figure 2.22.2.1: Glaciers in the Swiss Alps (center: Aletschgletscher) for 1996

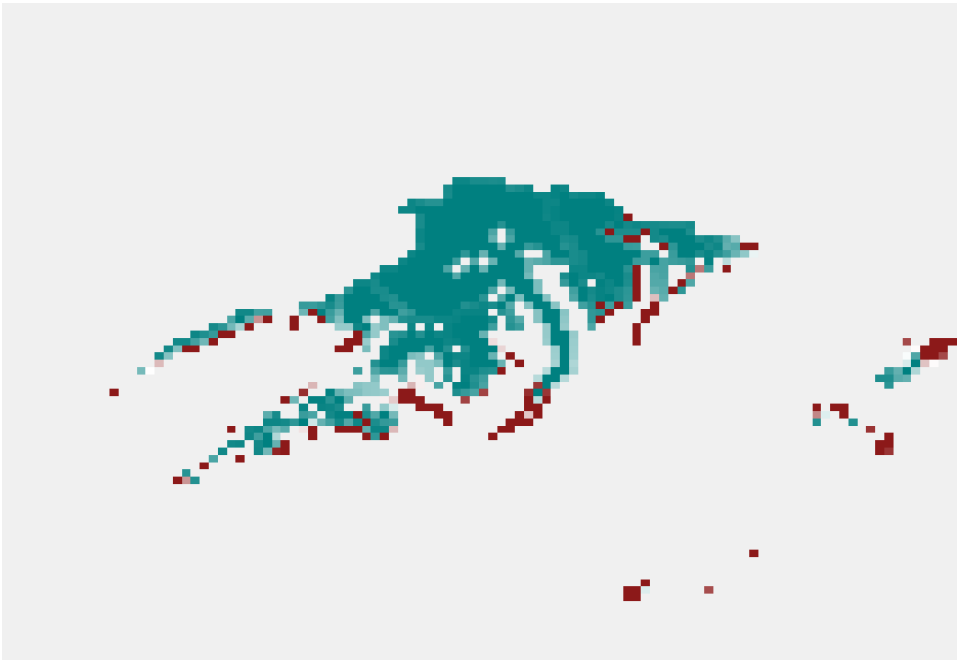


figure 2.22.2.2: Glaciers in the Swiss Alps for 2010

2.22.3 Per-Sub-Basin-Parameterization for most glacier model parameters

One interesting new feature of WaSiM 11.00.00 is the possibility to set almost all parameters of the glacier model as sub-basin-specific parameters. The mechanism in the control file works identical to the parameterization of the unsaturated zone model, i.e. all parameters can be defined for a range of subbasins and have to be provided table-like with the sub-basin codes as headers and the values per sub-basin code in the following lines.

Figure 2.22.1 shows the results of two WaSiM runs. The first run was done with equal settings for all glaciers. That run is represented by the orange and the green time series. The second run was done with the old parameters for all glaciers up to sub-basin code 34 and with changed parameters for all other glaciers (35 and higher). That run is represented by the purple and the blue line. As expected, the green and the blue line, stemming from different runs but having the same parameterization, have identical runoff values (i.e. sub-basin 34). But sub-basin 35 shows different values for the two runs, since sub-basin 35 has other parameters than sub-basin 34 in that second run, whereas sub-basin 34 had the same parameters in both runs.

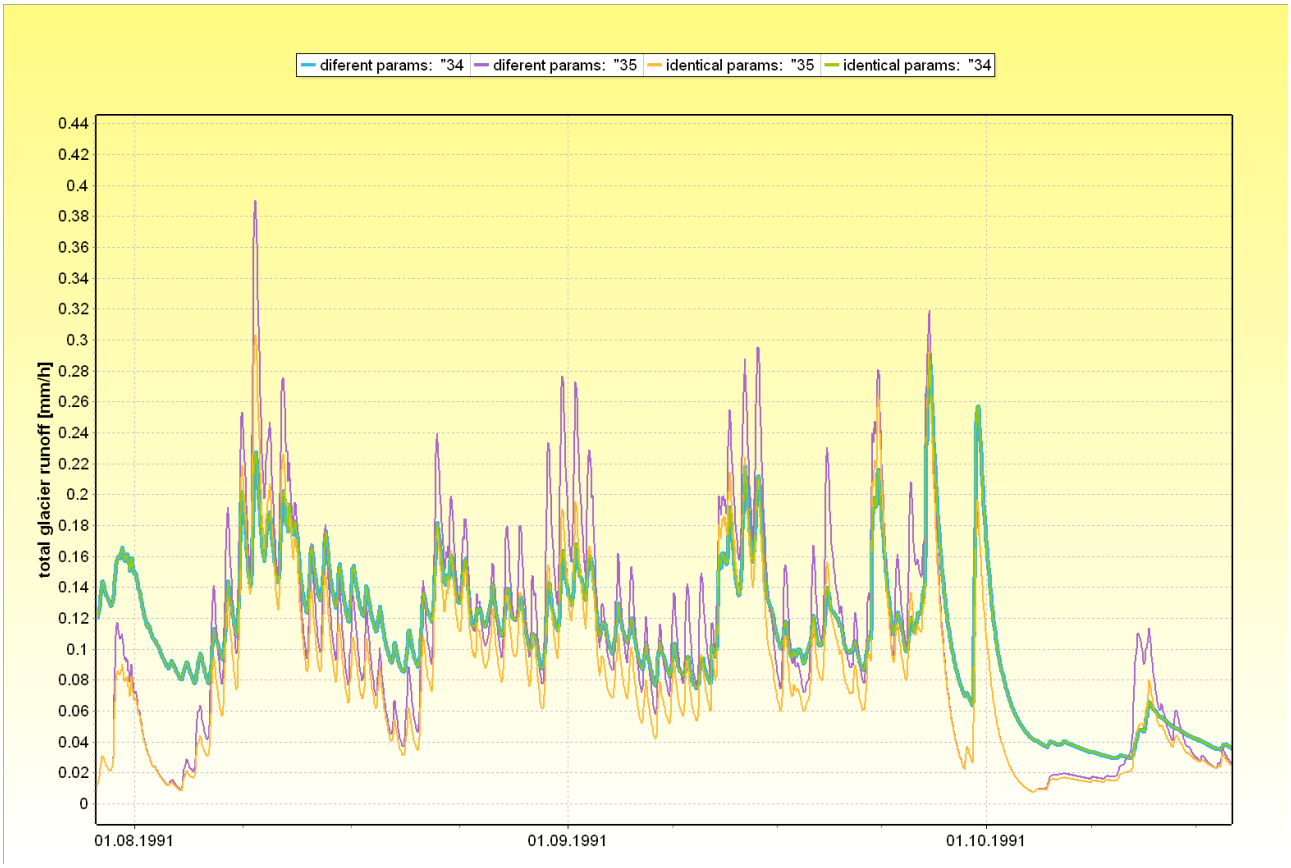


Abbildung 2.22.1: comparison of total glacier runoff for two sub-basins (glaciers) with a) identical parameters (green and blue) and b) different parameters (orange and purple)

The parameters that can be defined per subbasin are:

- (subbasin code must be the first line, contains the headers, i.e. the codes for the (range of) subbasins
- t-index factor for ice
- t-index factor for firn
- t-index factor for snow
- melt factor (when using radiation correction in methods 11...13)
- radiation coefficient for ice_min (for method 2/12/13)
- radiation coefficient for ice_max (for method 2/12/13)
- radiation coefficient for snow and firn min (for method 2/12/13)
- radiation coefficient for snow and firn max (for method 2/12/13)
- els-konstante for ice
- els-konstante for firn

- els-konstante for snow
- initial reservoir content for ice discharge (single linear storage approach)
- initial reservoir content for firn discharge (single linear storage approach)
- initial reservoir content for snow discharge (single linear storage approach)
- average Equilibrium line elevation in m (e.g. 2500)
- change rate of WE per m in mm (e.g. 2) -> every 100m the WE of firn in each layer will grow by 200mm

As one may notice also later in the control file example, the list of parameters contains 3 additional lines compared to the original WaSiM control file. To make it compatible with the old versions, the method code was extended by 100, i.e. the old ice_firn-method 13 and the new method 113 will both take the melt method from snow model, but method 113 expects the parameters per sub-basin (range) and also expects the additional header line and two additional lines for equilibrium line and change rate of WE with elevation (for cold start initialization only)

Control file sample:

All lines printed in bold are the new sub-basin specific parameters. The green marked lines are the additional lines for the sub-basin codes and for the firn creation parameters, which were a series of parameters in a single line like this :

7 **3200 1.8** # number of layers for the firn stack, followed by two initialization parameters: average Equilibrium line elevation in m (e.g. 2500) and change rate of WE per m in mm (e.g. 2) -> every 100m the WE of firn in each layer will grow by 200mm

Since such parameter-series are not compatible with a table-like structure, the new methods reads these two parameters as part of the new table structure and ignores everything after the first parameter in the above example.

```
[ice_firn]
113   # method for glacier melt: 1=classical t-index, 2=t-index with correction by
radiation, 11 = dynamic glacier model with classical t-index, 12 = dynamic glacier
model with radiation correction, 13 = melt methods are taken from snow model (for snow
only)
# > 100: same as method-100, but using per-sub-basin parameterization, with an
additional row for the subbasin code (as in unsatzonmodel) and for the dynamic glacier
model
# with per glacier/subbasin definable equilibrium line and WE change rate instead of
having those two values together with the number of firn layers in the same line
# 113 means: method is 13, but the following parameters may be provided per sub basin
with extra rows for the subbasin codes and for equilibrium line and WE change rated
1            34           35           123          # subbasin code
5.0000000    5.0000000    7.0000       7.0000       # t-index factor for ice
4.0000000    4.0000000    6.0000       6.0000       # t-index factor for firn
3.0000000    3.0000000    5.0000       5.0000       # t-index factor for snow
1.4000000    1.4000000    1.8000       1.8000       # melt factor (when using ra- ↵
#diaion correction in methods 11...13)
-0.0001000   -0.0001000   -0.0000660   -0.0000660   # radiation coefficient for ↵
#ice_min  (for method 2/12/13)
0.0003000    0.0003000    0.0005340    0.0005340    # radiation coefficient for ↵
#ice_max  (for method 2/12/13)
0.0000600    0.0000600    0.0000940    0.0000940    # radiation coefficient for snow ↵
#and firn min (for method 2/12/13)
0.0002500    0.0002500    0.0002840    0.0002840    # radiation coefficient for snow ↵
#and firn max (for method 2/12/13)
12.00000     12.00000     12.03400     12.03400     # els-konstante for ice
240.0000     240.0000     240.0340     240.0340     # els-konstante for firn
24.00000     24.00000     24.03400     24.03400     # els-konstante for snow
0.0100000    0.0100000    0.0100340    0.0100340    # initial reservoir content for ↵
#ice discharge (single linear storage approach)
0.0100000    0.0100000    0.0100340    0.0100340    # initial reservoir content for ↵
#firn discharge (single linear storage approach)
0.0100000    0.0100000    0.0100340    0.0100340    # initial reservoir content for ↵
#snow discharge (single linear storage approach)
3200.0000    3200.0000    2440.0000    2440.0000    # average Equilibrium line ↵
#elevation in m (e.g. 2500)
1.8000000    1.8000000    1.50000      1.50000      # change rate of WE per m in mm ↵
# (e.g. 2) -> every 100m the WE of firn in each layer will grow by 200mm
$outpath//$firn_melt # melt from firn
1//$Writegrid           # 0, if no grid-output is needed, else one of the codes
described above
$outpath//qfir//$grid//.//$code//$year $hour_mean # melt from firn as statistic file
$outpath//$ice_melt  # melt from ice
1//$Writegrid           # 0, if no grid-output is needed, else one of the codes
described above
$outpath//qice//$grid//.//$code//$year $hour_mean # melt from ice as statistic file
$outpath//qglc//$grid//.//$code//$year $hour_mean # melt from ice and firn as statistic
file
# ------------------------------------------------------------------------
# now some new parameters for the new dynamic glacier model (methods 11 and 12)
…
…
```

2.23 Dynamic phenology

Phenology is of growing importance for hydrological water balance simulations, especially in the context of impact studies on climate change. Recent studies have shown that phenological phases (i.e. start and length) can significantly change under changing climate conditions. In addition, phenology can also show pronounced year-to-year variations, mainly depending on the seasonal course of air temperature.

Until now plant development was statically simulated by using predefined intra-annual courses of

vegetation dynamics. Since WaSiM 7.8.x this deficiency is overcome. Now temperature-induced variations in the timing of phenological phases can be dynamically calculated. For this purpose, WaSiM offers three different phenological models which can only be used if time series of air temperature are available. The applicability of the models is not limited to daily model time steps but also possible in any other time steps. The necessary conversion from non-daily temperature values into daily values is performed internally by WaSiM.

Table 2.22.1: Comparison of phenological models implemented into WaSiM

Model name in WaSiM	**DynamicPhenology_ 1 (DP1)**	**DynamicPhenology_ 2 (DP2)**	**DynamicPhenology_ 3 (DP3)**	**DynamicPhenology_ 4 (DP4)**
Model type	„Thermal Time Model“	„Sequential Model“	„Thermal Time Model“	„Sequential Model“
Start of growing period	dynamically simulated, depending on forcing units only	dynamically simulated, depending on prior chilling amount and on forcing units	dynamically simulated, depending on forcing temperatures only (degree day sums)	Dynamically, simulated depending on prior chilling and on forcing units (similar to DP2)
Next stages of growing	predefined (relatively to the start day) – only the very next stage	predefined (relatively to the start day)	dynamically simulated, several stages possible	Dynamically for the very next stage and for leaf fall
Landuse type to simulate	trees, groves	trees, groves	grasses, arable crops	trees

2.23.1 Dynamic Phenology 1 (DP1)

DP1 represents a simply “thermal time model” and simulates the phenological response of trees to changes in air temperature. The model does not take dormancy into account. It assumes that the release of dormancy is usually reached at the end of the year. Leaf unfolding only starts when plant-specific heat sums (forcing units) have been reached. The “forcing units” are calculated after the logistic function of Sarvas (1974):

$$S_f(t)=\sum_{i=t_1}^{t} R_f(T_i) \qquad S_f(t_2):=F^* \qquad \textbf{(2.22.1)}$$

$$R_f(T_i)=0 \quad \text{for} \quad T_i \le T_{Bf} \qquad \textbf{(2.22.2)}$$

$$R_f(T_i)=\frac{28.4}{1+\exp\left(-0.185\cdot\left(T_i-T_{Bf}-18.4\right)\right)} \quad \text{for} \quad T_i > T_{Bf} \qquad \textbf{(2.22.3)}$$

with
- S_f state of forcing
- R_f forcing rates
- F^* forcing units; heat sum which has to be reached before the phenophase can be started
- T_i daily mean temperature [°C]
- T_{Bf} plant-specific base temperature for „forcing“ [°C]
- t_1 end of dormancy; start day for calculating F* [JulDays]
- t_2 start of the phenophase (active growth) [JulDays]

From Equations (2.22.1) to (2.22.3) it can be seen that DP1 only requires the optimization of three parameters: F^*, t_1 and T_{Bf} (in the WaSiM control file these parameters have slightly different names: F^*, *DP1_t1_dorm*, *DP1_T_Bf*).

The DP1 model can be activated by setting the plant-specific method identifier in the primary

[landuse_table] to “DynamicPhenology_1“. This table entry signalizes WaSiM that phenology has to be calculated dynamically and that DP1 parameters have to be read in.

Besides the three above-mentioned parameters WaSiM needs two additional parameter entries in its [landuse_table]:

- *JDReset_TStart* [JulDays]:
  guaranteed a proper start of a new vegetation period (reset from *TStart* to -1 and reset of the „forcing units“ to zero)
- *maxStartJDforDP1* [JulDays]:
  denotes the last possible starting day for the use of DP1 within the actual calendar year. In the case that the model is started after this date, *TStart* is set automatically to:

  $$TStart = maxStartJDforDP1 - delta \quad \textbf{(2.22.4)}$$

  Here *TStart* denotes the start of the phenophase (active growth) and has to be set to -1 in the parameter row *JulDays*. During the initialization the model looks for this value and stores the column number for *TStart*. The value in the following column is interpreted as delta [JulDays]. It denotes the end of leaf unfolding.

  It is worthy to note that the entries for *JulDays* before and after these both days (*TStart* and *delta*) are fixed (resp. variable with the altitude only) and used as always (cf. method “VariableDayCount”). That means that the course of the phenological development after its (dynamically calculated) start follows the well-known old model scheme and uses statically predefined vegetation curves. For these predefined sections it is possible to consider the influence of altitude on plant development whereas this feature can not be used for the column entries *TStart* and *delta*. In this case the parameter *AltDep* (altitude dependence) has to be set to zero because here the air temperature is already implicitly considered. Furthermore it is not possible to interpolate between parameters which are earlier than *TStart*.

```
4 some_vegetation { method = DynamicPhenology_1;
RootDistr        = 1.0;
TReduWet         = 0.95;
LimitReduWet     = 0.5;
HReduDry         = 3.45;
IntercepCap      = 0.3;
F*               = 175;
DP1_t1_dorm      = 60;
DP1_T_Bf         = 0;
JDReset_TStart   = 1;
maxStartJDforDP1 = 150;
JulDays          = 1     -1       +17    258    288    319    349 ;
...
```

2.23.2 Dynamic Phenology 2 (DP2)

DP2 is a „sequential model“ for simulating the timing of phenology for trees and groves. In contrast to DP1 it considers both the dormancy and the following phase of ontogenetic development.

Normally trees require a certain period with chilling temperatures before they can react to higher temperatures, which force the development of buds and induce leaf unfolding and lowering in spring (Chmielewski et al., 2005). That means that the release of dormancy is only possible if the “chilling units” are reached. The DP2 “chilling-forcing” model calculates the plant-specific heat and chilling units as follows:

$$S_c(t)=\sum_{i=t_0}^{t} R_c(T_i) \quad \text{with} \quad S_c(t_1):=C^* \qquad \textbf{(2.22.5)}$$

$$S_f(t)=\sum_{i=t_1}^{t} R_f(T_i) \quad \text{with} \quad S_f(t_2):=F^* \quad \text{and} \quad F^*=a\cdot e^{b\cdot C^*} \qquad \textbf{(2.22.6)}$$

$$R_c(T_i)=0 \quad \text{for} \quad T_i \le z_1 \quad \text{or} \quad T_i \ge z_2 \qquad \textbf{(2.22.7)}$$

$$R_c(T_i)=\frac{T_i-z_1}{T_{Bc}-z_1} \quad \text{for} \quad z_1 < T_i \le T_{Bc} \qquad \textbf{(2.22.8)}$$

$$R_c(T_i)=\frac{T_i-z_2}{T_{Bc}-z_2} \quad \text{for} \quad T_{Bc} < T_i < z_2 \qquad \textbf{(2.22.9)}$$

$$R_f(T_i)=0 \quad \text{for} \quad T_i \le T_{Bf} \qquad \textbf{(2.22.10)}$$

$$R_f(T_i)=\frac{28.4}{1+\exp\left(-0.185\cdot\left(T_i-T_{Bf}-18.4\right)\right)} \quad \text{for} \quad T_i > T_{Bf} \qquad \textbf{(2.22.11)}$$

with
- S_c state of chilling
- R_c chilling rates
- C^* chilling units; chilling sum which has to be reached before the dormancy can be finished
- S_f state of forcing
- R_f forcing rates
- F^* forcing units; heat sum which has to be reached before the phenophase can be started
- a, b empirical parameter
- T_i daily mean temperature [°C]
- T_{Bc} plant-specific base temperature for „chilling“ [°C]
- T_{Bf} plant-specific base temperature for „forcing“ [°C]
- z_1 constant (for forest = -3.4 and for fruit trees = 3.0)
- z_2 constant (for forest = 10.4 and for fruit trees = 12.0)
- t_o start of dormancy; start for calculating C* [JulDays]
- t_1 end of dormancy; start for calculating F* [JulDays]
- t_2 start of the phenophase (active growth) [JulDays]

Similar to DP1 the phenological model DP2 has to be activated by setting a plant-specific method identifier in the primary [landuse_table]. In the case of DP2 the table entry is “DynamicPhenology_2“.

For using DP2 WaSiM requires the following plant-specific parameter entries (for explanations see above):

```
# extract from the primary landuse table

[landuse_table]
1

5 oak  {method = DynamicPhenology_2 ;
RootDistr         = 1.0;
TReduWet          = 0.95;
LimitReduWet      = 0.5;
HReduDry          = 3.45;
IntercepCap       = 0.35;
StressFactorDynPhen = 1.0;
F*                = 221.2;
DP1_t1_dorm       = 1;
DP1_T_Bf          = 0;
DP2_t0_dorm       = 244;
DP2_t1_dorm       = 60;
DP2_T_Bf          = 0.2;
DP2_T_Bc          = 8.4;
DP2_Par_a         = 231.6;
DP2_Par_b         = 0.0;
DP2_Offset_1      = -3.4;
DP2_Offset_2      = 10.4;
JDReset_TStart    = 1;
```

```
maxStartJDforDP1 = 150;
JulDays          =  1            -1      +10      258            288            319            349;

...
```

Note that DP2 cannot be used for simulating the phenology of the first model year because at this time WaSiM does not know the state of chilling (S_c). In this case WaSiM changes automatically the phenological model from DP2 to DP1. This switch setting can only deactivated if results from previous model runs are available for the initialization of DP2.

2.23.3 Dynamic Phenology 3 (DP3)

The third implemented phenological model is DP3. It is a simple "thermal time model" focusing mainly on the simulation of phenophases for crops. DP3 calculates the forcing rates (R_f) very simplified as growing-degree-days (GDD) and not by means of the more complex Sarvas function (cf. DP1).

$$S_f(t)=\sum_{i=t_1}^{t} R_f(T_i) \qquad \text{with} \qquad S_f(t_2):=F^* \qquad \textbf{(2.22.12)}$$

$$R_f(T_i)=0 \qquad \text{for} \qquad T_i \le T_{Bf} \qquad \textbf{(2.22.13)}$$

$$R_f(T_i)=T_i-T_{Bf} \qquad \text{for} \qquad T_i > T_{Bf} \qquad \textbf{(2.22.14)}$$

with
S_f state of forcing
R_f forcing rates
F^* forcing units; heat sum which has to be reached before the phenophase can be started
T_i daily mean temperature
T_{Bf} plant-specific base temperature for „forcing"
t_1 start for calculating F^*
t_2 smallest whole number where $S_f(t_2) \ge F^*$ is valid; start of the phenophase

The phenological model DP3 is activated by setting the plant-specific method identifier in the extended [landuse_table] to "DynamicPhenology_3". Compared to the previous models (DP1 and DP2), DP3 does not only allow to simulate the starting day of active growth but also the timing of the subsequent phenophases.

The practical use of DP3 requires the definition of so-called "Sample-Days" which indicate the "maximum Julian days" where the actual phenophase should be started at the latest. The "Sample-Days" are plant-specific parameters. They have to be given within the parameter row *(max) JulDays* in the WaSiM [landuse_table].

The "Sample-Day" entries are closely linked to another newly introduced pheno-parameter, namely the "forcing thresholds". These thresholds (in the WaSiM control file named as *ForcingThreshold*) have to be defined separately for each "Sample-Day".

WaSiM interprets positive "Sample-Day" entries as threshold values which have to be reached by the "forcing rates" (R_f). In the case that R_f reaches the *ForcingThreshold* earlier than the corresponding "Sample-Day", the actual Julian day is used as start day of the new phenophase. In the other case (the *ForcingThreshold* is not reached earlier than the corresponding "Sample-Day") the given "Sample-Day" is taken as new start day.

It is important to know that the first and last "Sample-Day" entries cannot be calculated dynamically but have to be fixed to 1 and 366, respectively. In both cases, the corresponding *ForcingThresholds* have to be set to -1.

In general, "Sample-Days" are interpreted as fixed if the corresponding *ForcingThresholds* are set

to -1. Then plant growth can only be influenced by its altitude dependency.

WaSiM linearly interpolates between "Sample-Days". It can be discriminated between two scenarios:

Scenario 1: The next "Sample-Day" is fixed.

Here the temporal interpolation is done linearly by using the Julian-Day entries of the actual (fixed or dynamic) and next (fixed) "Sample-Days".

Example: The leaf area index (LAI) between day 60 (with LAI = 2) and day 100 (with LAI = 10) is interpolated to 4 at day = 70.

Scenario 2: The next "Sample-Day" is calculated dynamically.

Here the temporal interpolation considers both the reached "forcing rates" of the actual (fixed or dynamic) and the *ForcingThresholds* of the subsequent "Sample-Day".

Example: The LAI between day 60 (with S_f = 200 and LAI = 2) and the next (uncertain) day (with *ForcingThreshold* = 600 and LAI = 10) is interpolated to 4 at S_f = 300.

WaSiM does not allow using DP3 without having the parameter entries for DP1 and DP2. The simulated Julian days indicating the start and the end of the phenophases can be controlled by a data stack which is written at the end of the model run. The stack is labelled as "MultipleStartTimes1_End" and contains phenological information for each „Sample-Day". In addition the actual state of forcing (S_f) is saved in stack layer 0.

```
5 some_crop  {  method              = DynamicPhenology_3;
RootDistr        = 1.0;
TReduWet         = 0.95;
LimitReduWet     = 0.5;
HReduDry         = 3.45;
IntercepCap      = 0.35;
F*               = 175.2; # used for DynamicPhenology_1 and 2 and 3!:
DP1_t1_dorm      = 60;  # used for DynamicPhenology_1 and 2 and 3!:
DP1_T_Bf         = 0; # used for DynamicPhenology_1 and 2 and 3!
JDReset_TStart  = 1; # used for DynamicPhenology_1 and 2 and 3!:
maxStartJDforDP1 = 150; #  start  day  for  the  model  run  to  use  default
DynamicPhenology_1 as a fall back
(max) JulDays    =  1     110   140    258    288    319    366 ;
ForcingThreshold = -1     115   485    2300   -1     -1     -1  ;
Albedo           =  0.05  0.15  0.15   0.15   0.15   0.15   0.15;
rsc              =  10    60    60     60     80     90     90;
...
AltDep           =  0.0   0.0   0.0   -0.025 -0.025 -0.025 -0.025;
            }
```

2.23.4 Dynamic Phenology 4 (DP4)

This method also allows to estimate the start of the leave fall. A few additional parameters for land uses using this DP4-type are required in the control file. Required new parameters in the land use table are (with examples):

- method = DynamicPhenology_4; the new method code
- DP4_t0_dorm = 304; # November, 1st =304/305 # starting day (julian day number), chilling units will be summed up after this day of year
- DP4_t1_dorm = 32; # February 2nd =32 # starting day (julian day number), forcing units will be summed up after this day of year
- DP4_T0 = 9; # threshold temperatur for chilling units (if T < T0)

- DP4_T1 = 4; # threshold temperatur for forcing units (T > T1)
- DP4_Par_a = 1748; # tree specific parameter
- DP4_Par_b = -317; # tree specific parameter
- DP4_T_xylstop = 10; # threshold temperature, the moving average of temperature has to stay below for 5 days in order to start leave fall
- DP4_t_xs1 = 183; # time of year (julian day number), the moving average analysis starts from (for start leave fall estimation)
- DP4_SPSLF = 5; # (SamplePointStartLeaveFall) sample point ID which is replaced by the day of start leave fall with the original value as latest time

The following sequence from the control file shows the coding of the Julian day sample points:

```
JulDays   = 15    74    -1    +28   268   310   349 ;
Albedo    = 0.25  0.25  0.25  0.25  0.25  0.25  0.25;  # Albedo (snow free)
rsc       = 90    90    90    55    55    90    90;    # leaf surface resistance in s/m
AltDep    = 0.025 0.025 0.0   0.0   0.0   0.0  -0.025; # altidute dependency in days per
                                                        meter
```

...

The starting day for unfolding leaves is marked by a -1 in the JD-list. It is calculated using forcing units after:

$$T_S = \Sigma(T_{avg} - T_1) \quad \textbf{(2.22.15)}$$

For each day exceeding the temperature T_1 (which is DP4_T1), the T_S value is incremented by T_{avg} (the daily average of temperature) minus T_1. The critical value of T_S is calculated by

$$T_{S,crit} = a + b\ ln\ (C_D) \quad \textbf{(2.22.16)}$$

Here, *a* and *b* are the parameters DP4_Par_a and DP4_Par_b, respectively. C_D is the number of days between DP4_t0_dorm and DP4_t1_dorm when T_{avg} < DP4_T0.

The beginning of the leave fall in autumn is estimated by a moving average of temperature (7 days average). If this value is falling below DP4_T_xylstop for at least 5 subsequent days, the leave fall phase will start. This estimation starts only after DP4_t_xs1 was reached (so that no leave fall will take place in spring already when temperatures may be low, too).
To synchronize the Julian day table with the result of this moving average evaluation, the index in the JD-table for which the start leave fall date is valid must be given as parameter DP4_SPSLF. In the example above, this is 5. So the JD with value 268 will be replaced by the day calculated by the moving average analysis. However: the given value (of e.g. 268) is also a maximum allowed value. So, if the moving average of temperatures didn't reach the threshold of 5 subsequent days until the Julian day given in this table, then the JD-value there will be taken as start leave fall date and no moving average will be evaluated further.
The length of the next phenologic phase (leave fall) is taken from the difference between the JD-entry with index DP4_SPSLF and the next one (in the above example 310-268 = 42 days).

Note 1: Since DP4_t0_dorm is usually November, 1 st , the model will switch to DP1-Method when the model run starts after this day but before DP4_t1_dorm. Only the following years will be handled by DP4 then. So it's a good idea to start the first model year in November, already.

Note 2: It's a good idea to set altitude correction (AltDep) for the sample points which are dynamically calculated (the one with -1 and it's successor as well as the DP4_SPSLF and it's successor) to 0. Otherwise the model may behave unpredictable.

2.23.5 Considering the influence of soil moisture on phenology

A stress factor has been introduced into the [landuse_table] of WaSiM in order to simulate the

influence of soil moisture on phenological development. The factor is labelled as *StressFactorDynPhen* and can be defined separately for each landuse type. As shown in Equation (2.22.17) the stress factor is used as scaling parameter for the forcing units (F^*).

$$F^{*\prime} = F^* \cdot StressFactorDynPhen \cdot \left(\frac{\Psi(t) - HR\,eduDry}{150 - HR\,eduDry} \right) \qquad \textbf{(2.22.17)}$$

with F^* forcing units; heat sum which has to be reached before the actual phenophase can be started

ψ actual hydraulic head (calculated by using the van Genuchten parameters of the upper soil layer and the mean soil water content of the rooted soil zone)

HReduDry hydraulic head for beginning dryness stress (reduction of transpiration)

The stress factor can influence the plant development in two different ways: a) Stress values larger than 1 will probably accelerate the phenology, whereas a deceleration can be expected for values smaller than 1. No phenological stress occurs in the case of missing or negative stress factor values (the same is valid for *StressFactorDynPhen* = 0). A useful range for the stress factor is between 0.5 and 2.

2.23.6 Handling of phenological output data

The results of the phenological simulation can be saved and controlled by using the writegrid options in the section [variable_grids] of the WaSiM control file. The model allows creating outputs for the the following variables (outputs for each vegetation layer):

- SumOfForcingUnits (for DP1, DP2, DP3 and DP4)
- Pheno_start (for DP1, DP2, DP3 and DP4)
- SumOfChillingUnits (for DP2 and DP4)
- FStar_ForcingThreshold (for DP2 and DP3)

Furthermore, the above-mentionend grid outputs can also be used for the initialization of (serial) model runs (option $DPreadgrids with: 0 = initialize internally, 1 = read in from grid). In order to use the values of an old model run for initialization, the following grids should be read in as variable gruds (for each vegetation layer four grids, use numbering 2 to n for more than one layer).

```
# extract from section [variable_grids] for vegetation layer 1

$outpath//$forcingunitsgrid1     SumOfForcingUnits1               0 -1
$Writegrid
$DPreadgrids
$outpath//$TStartPhenoGrid1      Pheno_start1                     0 -1
$Writegrid
$DPreadgrids
$outpath//$chillingunitsgrid1    SumOfChillingUnits1              0 -1
$Writegrid
$DPreadgrids
$outpath//$FStargrid1                    FStar_ForcingThreshold1                  0 -1
$Writegrid
$DPreadgrids
```

The two codes behind the grid names have the following meaning:

The first code indicates if “nodata values” should be replaced automatically by “nearest neighbor values” (0 = no, 1 = yes). The second code is the default value which will be used if the grid will not be read from disk but created internally (see last parameter). For the phenological variables the latter code is set to -1 per default because this signalizes WaSiM that threshold values have not been reached yet.

After each grid, two more codes must be defined: the write cdode and the read code. The meaning is the same as for all other grids in the sections standard_grids or anywhere else (i.e. the write code has the same meaning as everywhere else). Note that here the variable $Dpreadgrids is used instead of $readgrids. This allows a more flexible way of setting the global values, but usually one will set

both variables to the same value.

2.24 Simple permafrost model

The simple permafrost model has been a temporary solution when the soil heat transfer model was not available. It is still available but not documented here because the heat transfer model is a much better choice for modeling permafrost impacts on hydrology.

2.25 Temperature and heat transport model

2.25.1 Theory

The following theoretical derivations were mainly obtained from Ronald Daanen, University of Alaska, Fairbanks, who supported the devel opment of the heat transfer model. The funding of the heat transfer model implementation was supported by the Environmental Sciences Division, U.S. Department of Energy, via the University of Alaska, Fairbanks (Arctic Next Generation Ecosystem Experiments “Arctic NGEE”).

The movement of heat in soils follows two principles: Conservation of heat (first law of thermodynamics) and Fouriers Law or the heat flux (second law of thermodynamics).

The conservation of heat means, that the change of the internal energy of a system is only due to energy added to or removed from the system and by work done by the system.

$$dU = \delta Q - \delta W \qquad \textbf{(2.24.1)}$$

where dU is the change in internal energy, δQ is the net balance of energy fluxes and δW is the work done by the system (all units in Joule). We can neglect the term δW, since we deal with a system consisting only of solid and liquid materials, and no external pressure changes are taken into account. Equation (2.24.1) can therefore be simplified into

$$dU = dQ \qquad \textbf{(2.24.2)}$$

The energy content of a soil can change over time due to temperature variations. The heat flux through the soil matrix (conduction) can be a large component of the ground heat transfer. Note that non-conductive heat transfer by flow of water and water vapour can be a dominant form of heat transfer in cold climates (Kane et al. 2001). The user can choose to simulate heat transfer by advection starting with WaSiM version 9.00.13 (October 2012).
For the process of heat transfer, the left and right side of Eq. (2.24.2) can be written as:

$$C(T)\cdot\frac{\partial T}{\partial t} = \nabla\cdot\lambda(T)\cdot\nabla T + q_s \qquad \textbf{(2.24.3)}$$

where $C(T)$ is the temperature dependent heat capacity of the soil , $\partial T/\partial t$ is the temperature change in time, $\lambda(T)$ is the temperature dependent thermal conductivity. The Nabla operator ∇ defines the temperature gradient (of the scalar field T) as ∇T and the divergence of the vector field $\lambda(T)\nabla T$ as $\nabla\cdot(\lambda(T)\nabla T)$; q_s is the source/sink term, i.e. heat that enters or leaves the system by advective transport due to water fluxes (later, that term will also be used for defining a lower boundary condition as known constant heat flux) . Eq. (2.24.3) could also be written as

$$C(T)\cdot\frac{\partial T}{\partial t} = \mathrm{div}\left(\lambda(T)\cdot grad\, T\right) + q_S \qquad \textbf{(2.24.4)}$$

(Note the resemblance to the water flux in the Richards approach!). This differential equation must be integrated to be used within the finite differences approach of the Richards equation. The discrete form of Eq. (2.24.4) applied to a 1D-vertical soil profile can be written as:

$$\rho\cdot V\cdot C(T)\cdot\frac{\Delta T}{\Delta t} = \lambda(T)\cdot\frac{A}{L}\cdot\Delta T = Q_{\mathrm{in}} - Q_{\mathrm{out}} + q_S \qquad \textbf{(2.24.5)}$$

where ρ is the density of liquid water (we will use 1000 Kg/m ^{3} here), V is the elementary volume (let's assume that it is 1 m ^{3}), A is the elementary area (let's assume as 1 m ^{2}), L is the elementary

length (we assume 1 m here), Q_{in} and Q_{out} are the heat fluxes through the upper and the lower boundary of the elementary volume.

Before breaking down the single terms in Eq. (2.24.5) , it should be noted that the heat transfer in a partly ice and/or water saturated soil (which means that there can be water, air and ice at the same time besides the soil matrix particles) is a little bit more complex than the heat transfer through a monolithic block of matter:

1. The phase change of water to ice and vice versa is a process depending on the soil porosity characteristics . Usually, the phase change mainly takes place between -10°C and 0°C, with water in the smallest pores freezing last (and thawing first) and water in the biggest pores freezing first (and melting last). In silty soils, the phase change reaches down to deeper temperatures (e.g. 90% ice, 10% water at -5°C possible, see Williams (1964)) than in sandy soils or even in peat (moor), where 90% of the water is already frozen at -0.5 to -1°C (Spaans (1996), Figure 1 and Nagare et al. (2011), Figure 5)
2. The thermal conductivity, λ , is also dependent on the ice or water content of the soil and is thus, indirectly dependent on temperature *T*. In fact, λ is an integrated parameter for the given soil's characteristics as the distribution of the individual components (soil matrix, water, ice, and air) should be taken into account.
3. The same is true for the heat capacity $C(T)$, where the soil matrix, ice, and water (we neglect air here) all have individual specific heat capacities. In addition, the phase change of water to ice, and vice versa, with its latent heat, L_h = 334 KJ/Kg, leads to changes in the effective heat capacity at any given temperature between 0°C and -10°C as, depending on the soil type, water can exist in soils below 0°C.
4. Changes is the distribution of water and ice, the effective thermal conductivity and the temperature dependent heat capacity (due to latent heat change) can be described using the same parameterization scheme as for the soil hydraulics properties: the van-Genuchten-parameterization. So the parameters *m*, *n* and α used for describing $k(\Theta)$ and $h(\Theta)$ can also be used in much the same way to describe the thermal properties.

Having said this, the following equations will describe the development of the soil heat transfer functions from its basic parameters. Note that the following algorithms will be applied iteratively for each numerical soil layer, side by side with the Richards-equation since both processes affect each other. Consequently, the soil heat transfer model is integrated into the soil model and could take a considerable amount of computing time depending on soil layer thickness heat gradients, thermal conductivity etc.

2.25.2 Estimation of the relative saturation, S_E

The first step in calculating 1D heat transfer by conduction is to estimate the fraction of ice and water, i.e. the relative saturation, S_E . A value of 1 means that there is only liquid water, while a value of 0 means that there is only ice. Note that an S_E value of 1 doesn't mean that the entire pore volume is filled with water. S_E only refers to the total amount of water that is present in the soil, which, in turn, is calculated by the mass transport equation following the Richard's approach.

$$\begin{aligned} S_E &= \left(\frac{1}{1+\left(\alpha \cdot \left| f \cdot T_{eff} \right|\right)^n} \right)^m \quad \text{for } T \le 0\,^\circ\text{C} \\ S_E &= 1 \quad \text{for } T > 0\,^\circ\text{C} \\ \text{with } T_{eff} &= \left(T + T_{Shift} \right) \end{aligned} \qquad \textbf{(2.24.6)}$$

where S_E is the relative fraction of liquid water of the total soil moisture; *n*, *m* and α are the van-Genuchten-Parameters as used in the Richards equation; *T* is the soil temperature in °C; and *f* is the solution of the Clapeyron-equation assuming zero ice pressure (333 KJ/Kg / 273.15 K at or near

0°C = 1.22 KJ/(Kg·K)). For temperatures above 0°C, S_E is always 1, whereas for temperatures below -10°C the value of S_E is assumed to be 0. The latter assumption is to reduce computational demand. Fine soils may still contain water below -10°C, but the fraction is negligible to any thermal and hydraulic transport process at the timescales of typical WaSiM applications.

The temperature shift expressed by T_{shift} accounts for unsaturated conditions by shifting the temperature dependence of S_E to lower temperatures, depending on the hydraulic pressure (which in turn depends on the relative water saturation of the pore volume). The algorithm to estimate the temperature shift is explained later when dealing with the latent heat change (see equation 2.24.10) and figure xxx 2.26.5.1).

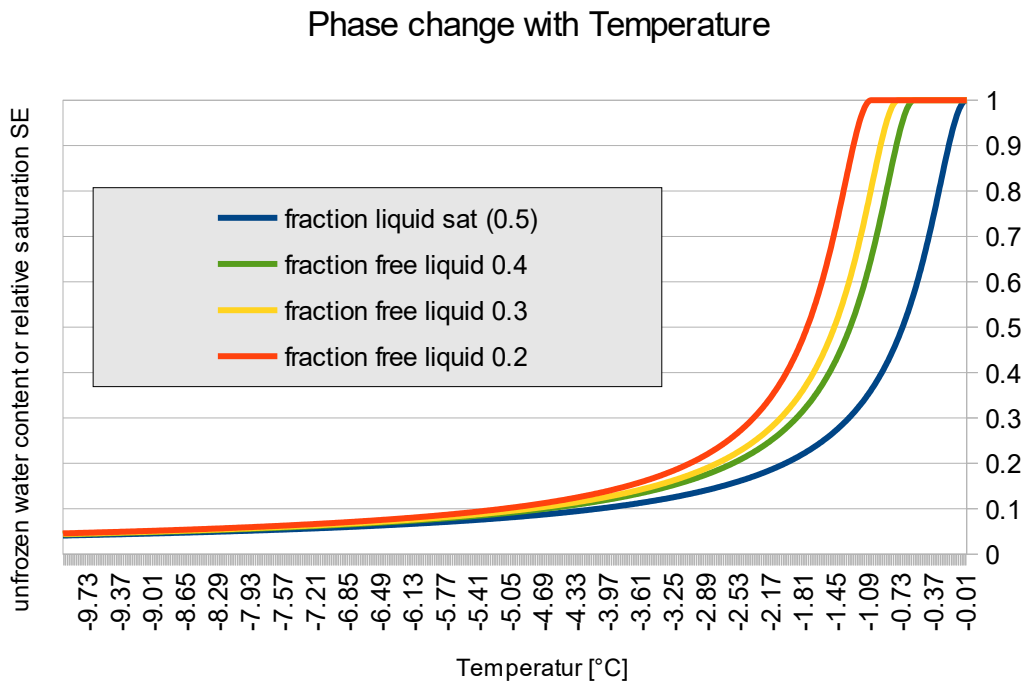


figure 2.25.2.1: Change in the fraction of liquid water, S E, during temperature change in a sandy loam soil (alpha=2, n=2, m=0.5).(temperatures are negative from -10 to 0°C)

Figure 2.25.2.1 shows the fraction of liquid water with temperature change computed for a silty soil (van-Genuchten parameters m =0.5 n =2 and α = 2, saturation water content 0.5, residual water content 0.1). The green, yellow and red curves show the relative saturation values for unsaturated soil with different water contents (see explanations for equation 2.24.10) below for details).

For a pore saturation value of Θ = 0.18 (which means a relative saturation or free liquid water content of 0.2 in terms of $\Theta_{rel} = (\Theta_{act} - \Theta_{res})/(\Theta_{sat} - \Theta_{res})$, red curve and arrow in the above figure), freezing starts at about -2.01°C only. This is exactly the temperature at which still 20% of the soil water are still unfrozen in a satura t ed freezing soil. So the red curve is shifted by -2.01°C to the left, reaching S_E =0.2 at -4.02°C. The main advantage off this approach is the ability of water to infiltrate in partly saturated soils at hydraulic cond u ctivities which are still as high as for unfrozen soil (see below).

2.25.3 Estimation of the effective thermal conductivity, $\lambda (T)$

The next step is to estimate the effective thermal conductivity which depends on the relative fractions of liquid water content, ice content (and thus on relative saturation S_E), dry soil and air content:

$$\lambda_{eff} = (\phi - \Theta)\cdot\lambda_a + (\Theta + \theta_{ds})\cdot\lambda_{ds}^{\theta_{ds}/(\Theta+\theta_{ds})}\cdot\lambda_l^{\theta_l/(\Theta+\theta_{ds})}\cdot\lambda_{ice}^{\theta_{ice}/(\Theta+\theta_{ds})} \qquad \textbf{(2 .24.7)}$$

where λ_{eff} is the effective thermal conductivity in J/(m·s·K); φ is the porosity, Θ is the total water content (ice and water), $\lambda_{a,\ ds,\ l,\ ice}$ is the thermal conductivity of air, dry soil, liquid water and ice, $\theta_{ds,\ l,\ ice}$ is the dry soil, liquid water and ice content as relative volume fractions. (the default value of λ_a is set to 0.0262 J/(m·s·K), λ_{ds} = 0.58 J/(m·s·K) - unless this parameter is defined in the soil table), λ_l = 0.5562 J/(m·s·K) and λ_{ice} = 2.33 J/(m·s·K). Note: θ_l can be expressed by $S_E \cdot \Theta$ (liquid fraction on water content or unfrozen soil water) and $\theta_{ice} = (1\text{-}S_E)\cdot\Theta$.

Note that we will make no difference whether the soil is completely saturated with water or not. All soils have a certain amount of soil water, and if there were no water at all in the soil, the connecting points between soil grains were so small, that the heat transfer will become extremely restricted – so the water improves the conduction massively, no matter if the soil is partly or fully saturated. The thermal conductivity of non-dry soil is thus much more like the conductivity of water or ice, so an effective value may be used here.

The effective thermal conductivity during temperature change (i.e. change in the fraction of liquid water) for a sandy loam soil is presented in figure 2.25.3.1. The van Genuchten parameters (*m, n* and *α)* used to calculate the fraction of liquid water are identical to those used in figure 2.25.2.1 with $\lambda_{soil,dry}$ = 0.58 J/(m·s·K), λ_{water} = 0.5562 J/(m·s·K) and λ_{ice} = 2.33 J/(m·s·K).

The different graphs show the effect of different total water contents Θ_{rel} (blue: saturated soil with *Θ*=0.5). As can be seen for saturated conditions (relative water content Θ_{rel} = 1), the thermal conductivity at 0°C (and warmer) is the weighted average of the dry soil thermal conductivity and the thermal conductivity of water, whereas the deeply frozen soil (near –10°C) has an almost constant value, which is mainly represented by the weighted average of dry soil and ice thermal conductivities.

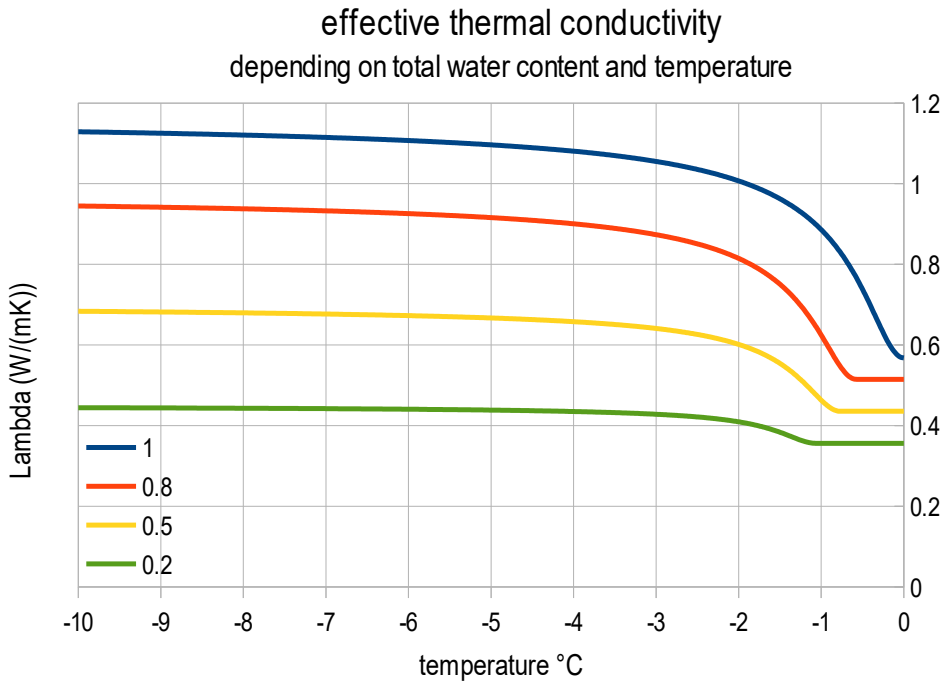


figure 2.25.3.1: Effective thermal conductivity plotted against soil temperature for different relative water contents Θ rel (1, 0.8, 0.5 and 0.2)

2.25.4 Estimation of the effective hydraulic conductivity

The effective hydraulic conductivity can be calculated using different approaches. The most simple approach would be to linearly interpolate between the saturated frozen and the saturated thawed hydraulic conductivities k_f and k_t:

$$k(T) = S_E \cdot k_t + (1 - S_E) \cdot k_f \qquad \textbf{(2.24.8a)}$$

where $k(T)$ is the effective hydraulic conductivity in m/s; k_t is the (thawed soil) hydraulic conductivity for $S_E = 1$ and k_f is the frozen hydraulic conductivity when all pores are filled with ice ($S_E \approx 0$). The frozen saturated hydraulic conductivity, k_f, can be defined as a parameter in the soil table (default is set to 10^{-8} m/s).

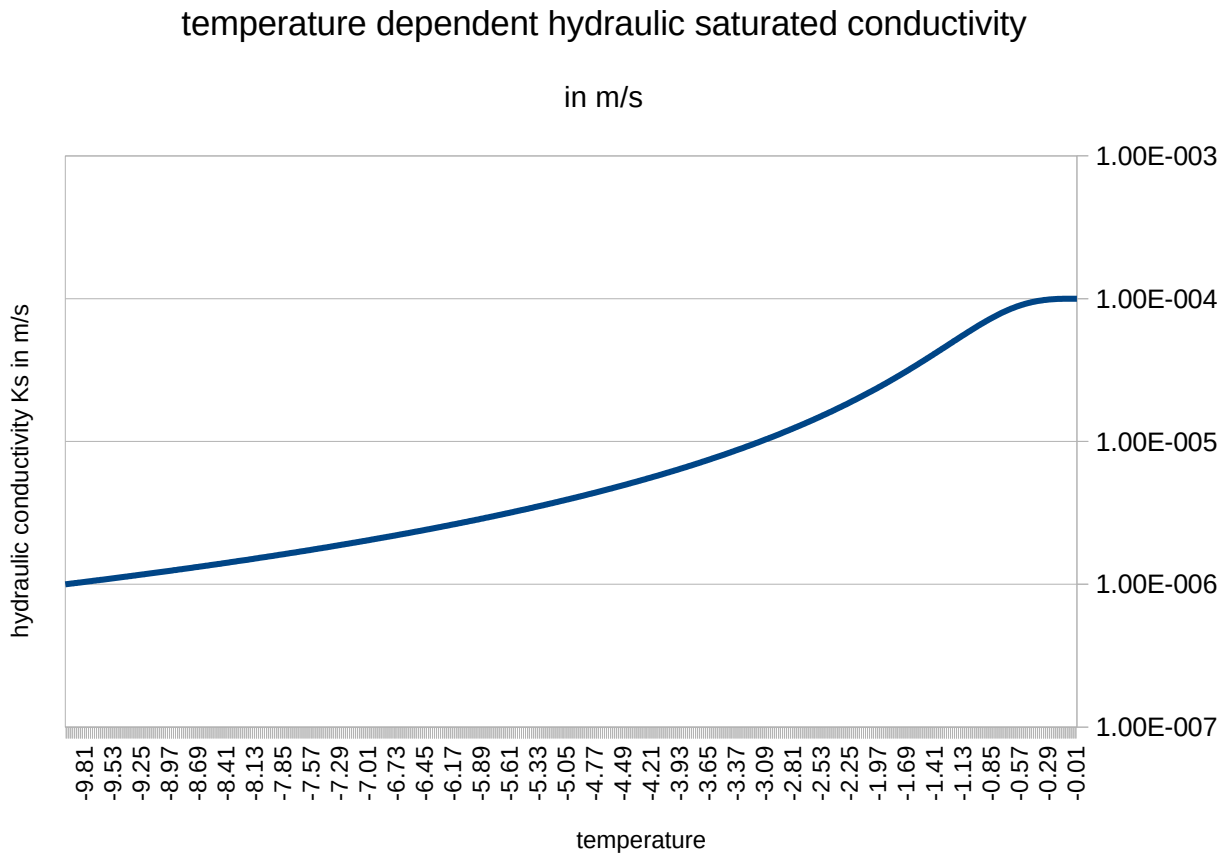


figure 2.25.4.1: Effective thermal conductivity based upon Eq. (2.24.8a)

The effective hydraulic conductivity decreases with an increasing fraction of ice, i.e. a decreasing S_E-value (Figure 2.25.4.1). The effective hydraulic conductivity changes with several orders of magnitude as the soil freezes/thaws. However, Eq. 2.24.8a does not result in the exponential response in effective hydraulic conductivity as has been measured for freezing/thawing mineral and organic soils (see Zhang et al. 2010). Accordingly, a modified exponential function could be applied to calculate the effective hydraulic conductivity (Figure 2.25.4.2):

$$k(T) = e^{S_E \cdot \ln(k_t(\Theta)) + (1 - S_E) \cdot \ln(k_f)} \qquad \textbf{(2.24.8b)}$$

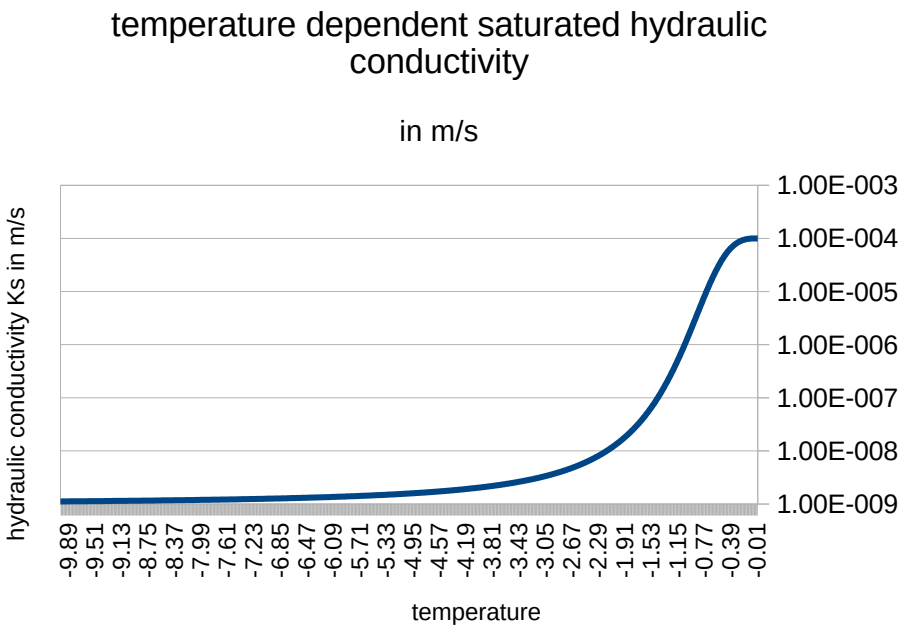


figure 2.25.4.2: Effective hydraulic conductivity using an exponential function (Eq. 2.24.8b).

The resulting graph (Figure 2.25.4.2) is more independent of the van-Genuchten α or n , since the error on the lower end of the temperature scale is now almost completely eliminated. This example uses a moderate frozen hydraulic conductivity of 10^{-9} m/s.
However, when soils freeze, they are not always saturated. Especially for peat, the hydraulic conductivity is not as small as equation (2.24.8b) suggests, since water can infiltrate into the largest pores quite good. The hydraulic conductivity must take into account all soil parameters as well as temperature dependent effects, e.g. by using the relative water saturation S_E.

Therefore, WaSiM uses an approach that again builds on the similarity between the parametrization of hydraulic and thermal conductivities in soils.

$$k(T, S_{min}) = k_{sat} \cdot k_{rel}(T, S_{min})$$
$$k_{rel}(T, S_{min}) = S_{min}^{0.5} \cdot \left(1 - \left(1 - S_{min}^{n}\right)^{m}\right)^{2} \qquad \textbf{(2.24.8c)}$$
$$\text{with } S_{min} = \min\left(S_E, S_R\right) \quad \text{and} \quad S_R = (\Theta_{sat} - \Theta_{res})/(\Theta_{act} - \Theta_{res})$$

Where $k(T,S_{min})$ is the effective hydraulic conductivity in m/s; k_{sat} is the saturated hydraulic conductivity in m/s; m, n and α are the van-Genuchten-parameters as used in the Richards equation, S_E and S_R are the unfrozen water fraction of the soil water and the filled relative pore volume, respectively; Θ_{sat} , Θ_{res} and Θ_{act} are the saturated, the residual and the actual water content of the soil, resp. This is basically the same as in equation (2.14.5) , except that here the relative free water content with respect to the pore volume can be replaced by the unfrozen water content for lower temperatures.

This approach results in specific unsaturated hydraulic conductivities for each saturation level, as is shown in figure 2.25.4.3. It compares the unfrozen water content of the soil S_E to the relative water saturation value from the Richards-equation set S_R. As long as the unfrozen water content of the soil S_E is higher than the value of S_R, the value of k_{rel} follows the van-Genuchten parameterization

of the Richards approach. When the soil freezes, S_E decreases until it eventually becomes smaller that S_R. Now, S_E is the limiting factor for the hydraulic conductivity.

The important thing about this approach is, that the infiltration into the uppermost unsaturated layer is always possible with a higher infiltration rate than given by the k-value (since otherwise a dry soil would never wet up). So once the uppermost layer contains some water, the hydraulic conductivity of that layer rises as well until it can supply some water to the lower next layer – as long as the water will not freeze and thus block the pores for more infiltrating water.

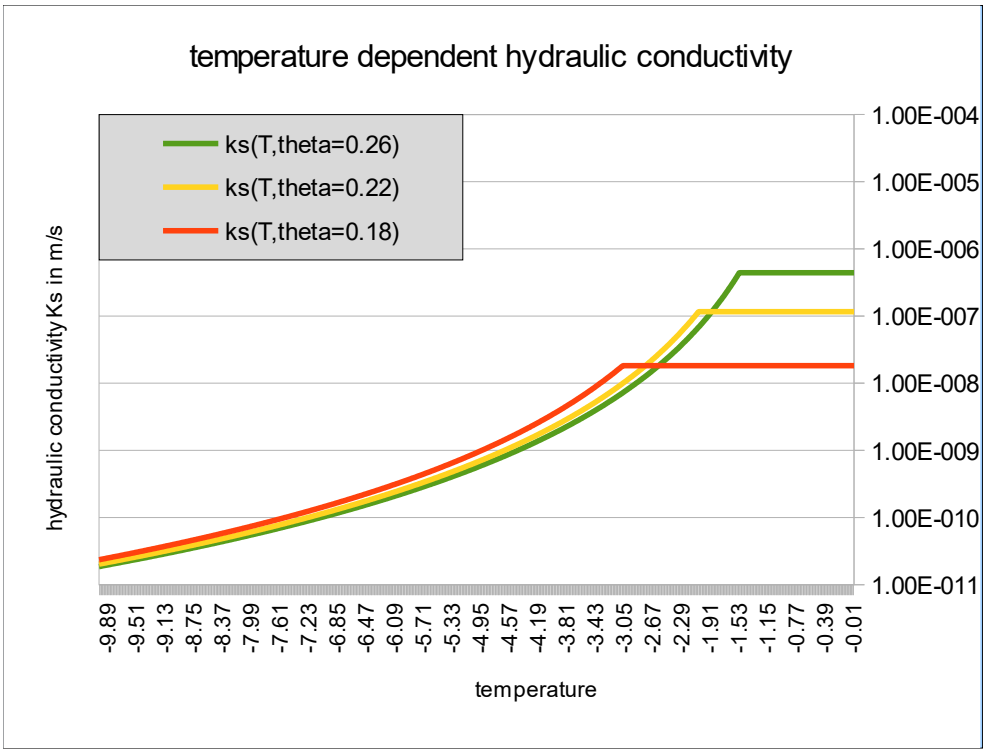


figure 2.25.4.3: Effective unsaturated hydraulic conductivity using an exponential function and a variable ks-value based on equation (2.24.8c)

Figure 2.25.4.3 shows a shift of the temperature dependent hydraulic conductivities to slightly higher values for soils with less total water (filled pores). This is an effect of the higher fraction of liquid water (unfrozen water S_E), since the water in the smallest pores freezes last and thus there may still be some minor water flux possible, although the hydraulic conductivity is very soon reaching extremely low values with falling temperatures.

Note: the example in figure 2.25.4.3 uses the same van-Genuchten parameters (n=2, α=2 m^{-1}) as in figure 2.25.4.1 and that no parameter is needed for the frozen conductivity as was the case for the other methods above.

2.25.5 Estimation of the effective heat capacity, *C*(*T*)

The heat capacity, $C(T)$, is by definition the amount of energy per unit volume (m^3) required to raise the temperature by 1 K. The effective heat capacity can be calculated by knowledge of the volume fractions of soil (mineral or organic), water, ice, and air, and their respective specific heat capacity and density (we neglect the air here):

$$C(T) = \rho_{soil} \cdot C_{\text{soil,dry}} + \rho_l \cdot \Theta_{\text{act}} \cdot \left(S_E \cdot C_t + (1 - S_E) \cdot C_f \right) + \frac{\partial E}{\partial T} \qquad \textbf{(2.24.9)}$$

where $C(T)$ is the effective heat capacity in J/(m^3·K),

C_t the specific heat capacity of liquid water is a material "constant" given by 4187 J/(Kg·K),

C_f the specific heat capacity of ice is given by about 2000 J/(Kg·K) (depending on temperature between 1940 J/(Kg·K) at -20 °C and 2090 J/(Kg·K) at 0 °C),

$C_{soil,dry}$ the specific heat capacitiy of dry soil, which is a material constant in J/(Kg·K) that must be defined as a parameter in the soil table (default = 800 J/(Kg·K)),

ρ_{soil} the dry density of soil in kg/m ^{3} (must also be defined in the soil table, default = 1500 Kg/m ^{3})

ρ_l density of (liquid) water at 0 °C ≈ 1000 Kg/m ^{3}

Θ_{act} actual water content in m ^{3} /m ^{3} following the Richards-approach

The energy change $\partial E / \partial T$ (in J/(m ^{3} ·K)) can be described using the van-Genuchten-parameters:

$$\frac{\partial E}{\partial T} = L_f \cdot \rho_l \cdot (\Theta_{act} - \Theta_r) \cdot m \cdot n \cdot \alpha \cdot f \cdot \left(-\alpha \cdot f \cdot T_{eff}\right)^{(n-1)} \cdot \left(1 + \left(-\alpha \cdot f \cdot T_{eff}\right)^n\right)^{(-m-1)} \quad \text{in } \frac{\text{J}}{\text{m}^3 \cdot \text{K}} \qquad \textbf{(2.24.10)}$$

$$\text{with } T_{eff} = \left(T + T_{shift}\right) \quad \text{for } T < 0\,^\circ\text{C}$$

where L_f latent heat of fusion for ice = 334000 J/Kg

ρ_l density of (liquid) water at 0 °C ≈ 1000 Kg/m ^{3}

Θ_{act} actual water content in m ^{3} /m ^{3}

Θ_r residual water content in m ^{3} /m ^{3}

m,n,α van-Genuchten parameter as used in the soil table

f factor connected to the solution of the Clapeyron-equation in m/K. Literature suggests different values (123 m/K in Daanen and Nieber (2009), (-)1.8 m/K in Grant (2000). We will use rather values in the range of 1.22 or 1.8 m/K, since higher values shift the freezing point even for soils with a very small fraction of liquid water extremely near to the 0 °C line (e.g. -0.02 °C for f = 123)

T Temperature of the modelled volume (assuming even distribution for water, soil matrix, ice and air in a numeric layer)

T_{eff} The shift in temperature due to partly unsaturated conditions, depending on the actual volumetric water content

T_{shift} Temperature shift for starting freezing due to higher pore pressure for unsaturated conditions

Figure 2.25.5.1 shows the change in energy (latent heat) with a given change in temperature (note that δ E/ δT = 0 for T > 0°C). The graph represents a differential expression so the resolution of the x-axis (0.01 K) has to be taken into account. If the temperature of a volume of ground changes by 0.01 K, then the change in energy of that volume is presented by the value on the y-axis. Energy is lost from the volume if the ground freezes, and gained by the volume if the ground thaws (ice melt). For example, for a temperature change from -0.5 °C to -0.51 °C (freezing), the amount of energy released is 997060 J/m ^{3} (Θ_s = 0.5 and Θ_r = 0.1, n=2, α =2). The entire area below the graph, which is the total energy that would be required to thaw the soil, should result in a latent heat of $L_f \cdot \rho_l \cdot (\Theta_s - \Theta_r)$ = 334 KJ/Kg · 1000 Kg/m3 · (0.5 – 0.1) = 133.6 MJ/m ^{3} for saturated conditions (blue graph), whereas the area below red graph has a reduced value that corresponds to the water content, in this case 0.2 · 133.6 MJ/m ^{3} = 26.72 MJ/m ^{3} .

Note that for saturated soils the integral of the part of equation 2.24.10) after the term (Θ_{act}-Θ_r) will give the result of 1.0 for ideal conditions (with integration limits from -∞ to 0), but practical applications (with integration limits from -10°C to 0°C) will usually result in slightly smaller values like 0.99 or so (for clay this may even be considerably less at about 0.6 only).

The change in effective heat capacity is at its largest when the change in water/ice fraction is high (slope of the phase change diagram is steepest), which coincides with the largest latent heat change (Figure 2.25.5.2). Water has a higher heat capacity than ice. The maximum slope of phase change and the maximum change in latent heat is at about -0.57 °C, whereas the maximum heat capacity is already at approx. -0.25°C for this specific soil, since the frozen part of the soil water reduces the overall heat capacity due to its lower heat capacity.

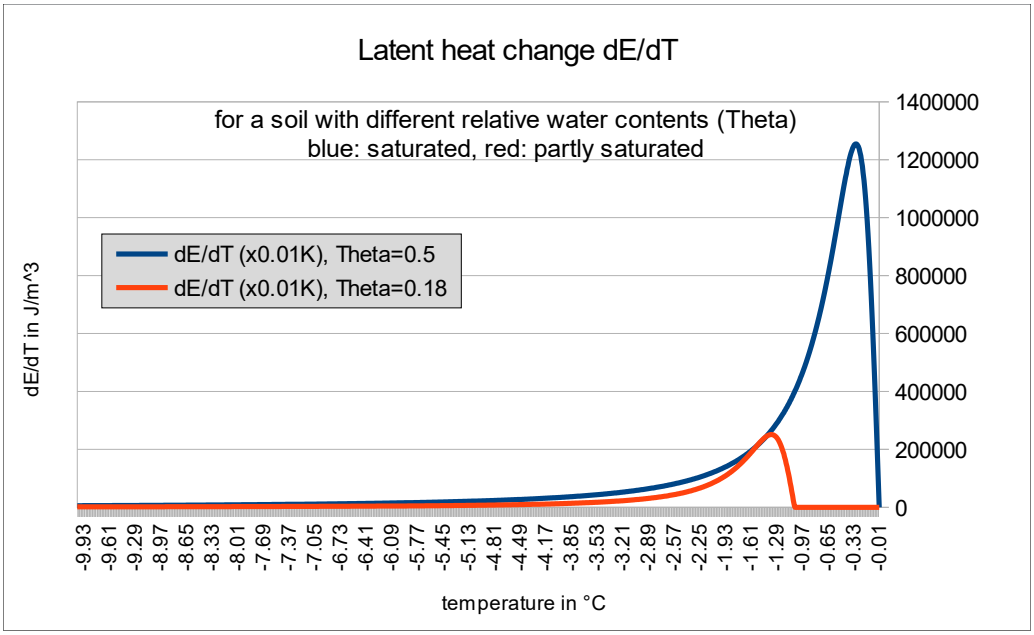


figure 2.25.5.1: The change in energy, i.e. latent heat, per temperature change of 0.01°C for a soil with Θ s = 0.5 and Θ r = 0.1 for saturated (Theta=0.5) and unsaturated (Theta=0.18) conditions.

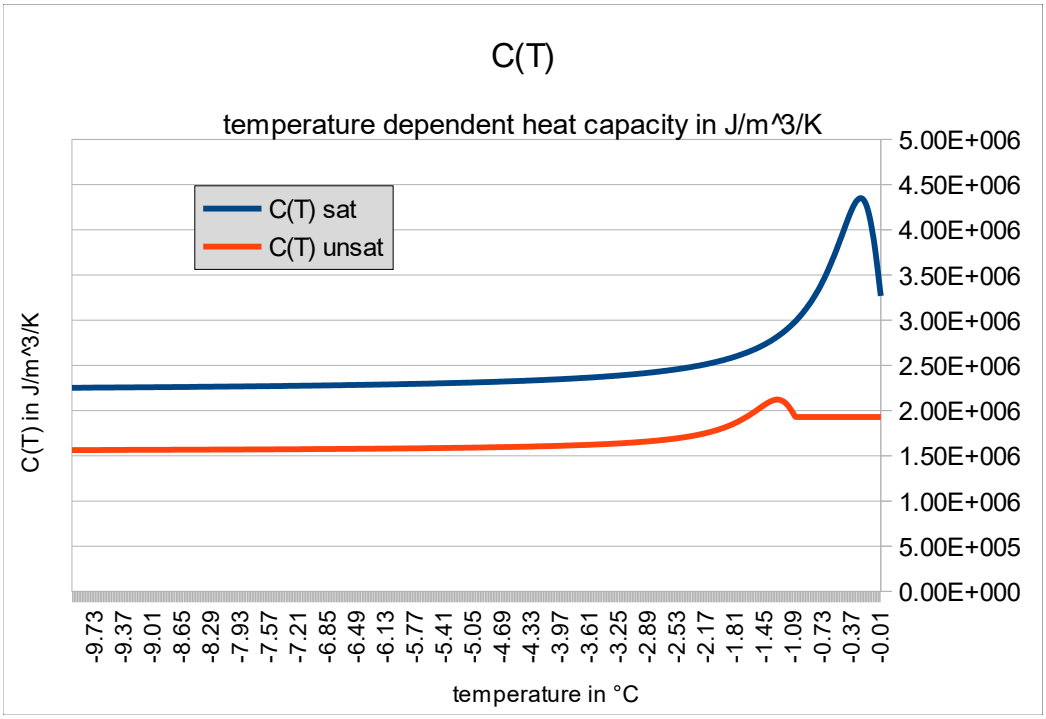


figure 2.25.5.2: The change in effective heat capacity of a freezing/thawing soil (Θ s = 0.5 and Θ r = 0.1) for saturated (Theta=0.5 and unsaturated (Theta=0.18) conditions.

2.25.6 Estimation of the temperature shift for unsaturated soils

Figure 2.25.5.1 shows the energy change with temperature for a saturated soil (blue) and an unsaturated soil (red). The red curve is reduced according to the relative free water content (liquid or frozen) of the soil (which is 0.2 in this example) and shifted to lower temperatures in a way that there should be no higher energy changes for any temperatures than for the same temperature in a saturated soil. This constraint is a result of considering the physics of freezing soils. Since equation 2.24.10) is quite complex for an analytical solution of the derivative, the steps described below will be done numerically by using lookup-tables in the model to speed up the program. The aim is to shift the reduced curves for unsaturated soils in a way that the maximum of the reduced energy change curve comes as near to the left branch of dE/dT as possible without letting the reduced energy change curve cross the saturated energy change curve (slight overshooting is allowed to simplify the shifting procedure).

Step 1: Finding the maximum energy change $\max(\mathrm{d}E/\mathrm{d}T)_{sat}$ and the corresponding temperature $T_{\mathrm{maxd}\ E,sat}$ for saturated soils. This could be done by setting the first derivative, which is the second derivative of E itself, to zero and solving for T. Since this is analytically not feasible (the resulting equation is some rows long...), it is done by numerical interpolation between the values of a lookup table containing the derivatives of dE/dT.

Step 2: Calculating the reduced maximum for unsaturated soils which is $\max(\mathrm{d}E/\mathrm{d}T)_{sat} \cdot \Theta_{rel}$.

Step 3: Calculating the corresponding temperature for that energy change for saturated soils $T_{maxdE,unsat}$.

Step 4: The difference between the corresponding temperatures for the maximum energy change for unsaturated soil and the energy change in saturated soils corresponding to the maximum for unsaturated soil $T_{\mathrm{maxd}\ E,sat} - T_{maxdE,unsat}$ is the basic temperature shift.

Step 5: The maximum of the reduced curve should be shifted a little bit to the right of the point calculated in steps two and three, so an empirically found correction factor of f_c = 1.15 (a range of 1.1...1.2 seems to be possible depending on main soil type, but 1.15 works good for a wide range of soils) is applied to the temperature shift from step four since this was computed as temperature difference between the peaks of saturated and unsaturated energy change, but we want the reduced curve to meet the unsaturated curve as asymptotically as possible at the left decreasing branch.

The effective temperature as used in equations (2.24.6) and 2.24.10) can such be expressed by

$$T_{eff} = T - \left(T_{max(dE/dT),unsat} - T_{max(dE/dT),sat} \cdot f_c \right) \qquad \textbf{(2.24.10a)}$$

in order to reduce the additional shift by factor f_c for saturation values near full saturation, factor f_c is reduced linearly to zero as soon as the temperature difference for the maxima of saturated and unsaturated soils get smaller than the double offset:

$$f_c = 1.15 \quad \text{for} \quad T_{max(dE/dT),unsat} < T_{max(dE/dT),sat} \cdot \left(1 + 2 \cdot \left(f_c - 1\right)\right)$$
$$f_c = 1 + 0.5 \cdot \frac{T_{Max(dE/dT),unsat} - T_{max(dE/dT),sat}}{T_{max(dE/dT),sat}} \quad \text{for all higher} \quad T_{max(dE/dT),unsat} \qquad \textbf{(2.24.10b)}$$

If the water content arrives saturation, f_c is going to 1.0, so the temperature shift is zero at this po i nt as should be e x pected.

2.25.7 Modeling the temperature change

At this point, the above equations (2.24.6) to 2.24.10) can be used to calculate temperature changes in a given soil volume. Equation (2.24.5) is repeated here for clarity:

$$\rho \cdot V \cdot C(T) \cdot \frac{\Delta T}{\Delta t} = \lambda(T) \cdot \frac{A}{L} \cdot \Delta T = Q_{in} - Q_{out} + q_s \qquad \textbf{(2.24.11)}$$

Eq. 2.24.11 (or eq. 2.24.5) is the discrete form of eq. (2.24.3) The heat transfer model (conduction and advection) described here for the WaSiM soil model is 1D vertical only. The model domain is presented in figure 2.25.7.1.

The layer of interest is the layer with index i. The layer above is layer i-1, the layer below is layer i+1. Q_{in} in equation (2.24.11) is the heat flow from layer i-1 to layer i (upper arrow), Q_{out} is the heat flow from layer i to layer i+1 (lower arrow). Each numerical layer may have its specific thickness d, but the effective length for heat transfer is calculated from center to center: for Q_{in}, the effective length is L_{i-1}, the effective length for Q_{out} is consequently L_{i+1}.
For clarity, the following heat flux equation refers to an elementary area A_e of 1m^2.

All variables of the left side of equation (2.24.11) can have specific values for each layer (ρ will usually be identical, but V and $C(T)$ may change). Parameters like λ, C, S_E and others will have to be averaged based upon the properties of both involved layers that represent the domain for fluxes Q_{in} and Q_{out}.

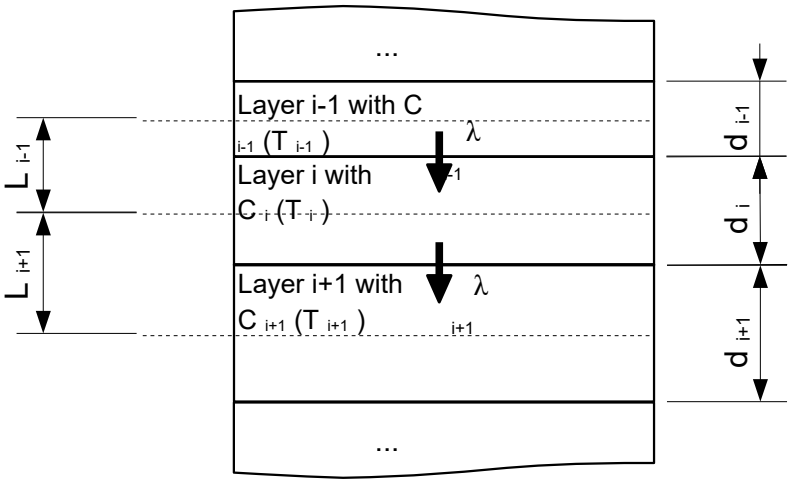


figure 2.25.7.1: Definition of layer indices for the discrete 1D heat transfer model describing heat transfer through conduction.

Having defined the model domain for the discrete heat flux model, the fluxes Q_{in} and Q_{out} can be written as:

$$Q_{in} = \lambda_{i-1}(T_{i-1}) \cdot \frac{A_e}{L_{i-1}} \cdot (T_{i-1} - T_i) \qquad [\mathrm{J \cdot s^{-1}}] \qquad \textbf{(2.24.12)}$$

and

$$Q_{out} = \lambda_{i+1}(T_{i+1}) \cdot \frac{A_e}{L_{i+1}} \cdot (T_i - T_{i+1}) \qquad [\mathrm{J \cdot s^{-1}}] \qquad \textbf{(2.24.13)}$$

Note that the values of $\lambda_{i-1}(T_{i-1})$ and $\lambda_{i+1}(T_{i+1})$ are **not** the values valid for their respective layers but effective values for the distance or volume between the centers of the involved layers.

The equation for the temperature change is achieved by using (2.24.12) and (2.24.13) for the respective terms in (2.24.11) and isolating ΔT on the left side:

$$\Delta T_i = \frac{\Delta t}{\rho \cdot V \cdot C(T_i)} \cdot \left[\lambda_{i-1}(T_{i-1}) \cdot \frac{A_e}{L_{i-1}} \cdot (T_{i-1} - T_i) - \lambda_{i+1}(T_{i+1}) \cdot \frac{A_e}{L_{i+1}} \cdot (T_i - T_{i+1}) + q_s \right] \qquad \textbf{(2.24.14)}$$

Eq. (2.24.14) allows a calculation of the change in temperature for each numerical soil layer (since this is an explicit solution which is supported in WaSiM for conduction only, the term q_s is not regarded specifically here; the implicit solution schema however is using additional sources and sinks, see below).

2.25.7.1 Boundary conditions

The finite difference scheme needs to have an upper and a lower boundary condition. The **upper boundary condition** will be the air temperature modified by an n-factor (Kade et al. 2006) as no physically based heat transfer modelling of snow cover is yet implemented in WaSiM. The n-factor, the ratio of seasonal thawing or freezing degree-day sums at the soil surface to that in the air, integrates the effects of all surface factors of the soil thermal regime and is the ratio of soil surface temperature to air temperature. In WaSiM, the n-factor can be defined in the control file with mean monthly values. A factor of 1 means that the air temperature is taken without scaling as upper boundary condition. Lower n-factors will lead to colder soil surface than air temperature in summer and warmer soil surface temperatures than air temperatures in winter.

There are two methods to define the lower boundary condition: A constant temperature or a constant

heat flux. The latter is recommended for long-term simulations (>10 yrs).
The constant temperature condition can be defined either globally or as a grid value for each cell.

1) Global definition (is used if no grid with identifier " `T_Lower_Boundary_Condition` " is found): The default value for the mean annual temperature, normalized to sea level is defined together with a temperature lapse rate (e.g. -0.7 K/100m). These values are then used to estimate the temperature at the lower soil boundary. For example, a meteorological station in the watershed is located at 350 m elevation and has a mean annual air temperature of 5.55 °C. Here, the normalized sea level air temperature would be 8 °C (calculated from 5.55+3.5*0.7). This would of course result in a lower boundary condition at the meteorological station of 5.55°C (8-3.5*0.7). Accordingly, at another location in the watershed, let's say with a ground surface elevation of 700 m, the lower boundary condition would be 0.65°C (from 5.55-7*0.7). This scheme is applied at start-up for each cell.
2) As a pre-defined Grid: An additional standard grid will be read in, containing the temperature at the bottom of the soil column (e.g. 5.55 for the given grid cell from the example above). The grid must be assigned the internal identifier "`T_Lower_Boundary_Condition`" in the control file. A pre-defined grid allow the user to custom design the lower boundary conditions based on for example, differences in air temperature and variables affecting local n-factors, which can dramatically modify mean annual soil surface temperatures (and therefore the lower boundary condition).

If the **lower boundary condition** should be defined as a **constant heat flux**, an *additional* boundary condition grid needs to be identified as "`HeatFlux_Lower_Boundary_Condition`". The constant heat flux is applied by calculating the resulting temperature difference between the lower boundary and the lowest numerical layer for the given heat flux. The resulting temperature is then set as lower boundary condition value ("constant" temperature) for every time step. Note that an T_Lower_Boundary_Condition–grid is still needed for this method for initialization purposes (see below). The measure of the grid cells for the constant heat flux is in W/m^2. Typical values are $65mW/m^2$ (world-wide average), so the value in the grid should be around 0.065. There exist some maps for the geothermal gradient (which is often expressed in W/m^2). Values range from $<40mW/m^2$ to around $1W/m^2$ and more, especially in geologically active regions.

Initial values of soil temperatures conditions:

The most basic initialization of the soil temperature profile in WaSiM assumes a linear gradient between the lower boundary condition and the upper boundary condition (air temperature modified by the n-factor) as provided by the forcing data that is informing the first model time step.
This means, that the user still has to define the lower boundary temperature via the `T_Lower_Boundary_Condition`-grid even if the constant heat flux method has been chosen. Note that the `T_Lower_Boundary_Condition`-grid or the default global boundary temperature settings are only used at the very first model time step if the boundary condition method is set to use the constant heat flux (i.e. if a `HeatFlux_Lower_Boundary_Condition-grid` is found). This implies that the initial conditions strongly depend on the model start date, since the upper boundary conditions (air temperature and n-factor) experiences seasonality.

When starting WaSiM with initialization from formerly stored grids, the temperatures are read in as a grid-stack (like soil water content etc.) and the model run can start with the correct (modeled) soil temperatures (no initializing from the boundary conditions occurs).

2.25.7.2 How does the numerical scheme work?

For the numeric solution, the various calculation steps are arranged as follows:

1. computing the length of the sub time step for the heat transfer model (by using the so called mesh Fourier number when using an explicit solution scheme or by the manually defined sub time step limits) and the sub time step for the Richards solution scheme (heat transfer sub time step will always be smaller or equal to the Richards sub time step)
2. start of a sub time step
3. estimating the relative (water to ice) saturation S_E
4. estimating the hydraulic conductivity using the van-Genuchten parameters and Eq. (2.24.8c) from the actual water saturation, Θ_{act} and the actual relative Saturations, S_E and S_R .
5. estimating the effective thermal conductivities
6. estimating the effective heat capacities
7. calculating the heat fluxes, the temperature differences and finally the resulting new temperatures using either explicit or implicit solution schemes (when using implicit schemes, only the new temperatures are calculated directly)
8. checking for breaks of the second law of thermodynamics (i.e. no source region may cool down to temperatures below the temperature of the target region) – restrict the fluxes if necessary or set a smaller time step. This is a security measure against numerical instabilities when using the explicit solution method
9. repeat heat transfer computation if necessary (depending on the ratio of the sub time step for heat transfer and the sub time step for soil water balance) --> go to step 3
10. calculating mass fluxes using the Richards approach
11. run the next sub time step for soil water balance

Note: Since the heat transfer is not limited to the unsaturated zone of the soil, the model domain of the Richards-Equation and the model domain of the heat transfer equation will differ: The Richards-approach is only applied to the unsaturated (numeric) soil layers, whereas the heat transfer must be calculated for each layer down to the lower boundary condition, i.e. for the entire soil column. In order to keep the computation speed as fast as possible, both algorithms will be processed separately but in the same main iteration loop (for sub time steps of the soil water balance): Each equation is applied to the respective model domain before calling the other equation or the next sub time step.

Some additional features were implemented into the model to optimize the performance. These are mainly a sub time step optimizer, and a scheme preventing the heat transfer model from inverting gradients when the sub time step was chosen too long (for explicit solution schemes):

1) Optimizing sub time steps: Even when WaSiM runs in hourly time steps, the duration of an interval is much too long to realistically calculate the heat flow into or between thin soil layers. So the internal time step has to be set to much smaller values down to a few seconds (depending on several parameters like layer thickness, soil water content, freezing state and temperature gradient and van-Genuchten parameters). On the other hand, if the temperature gradients are small, the time step can be as large as some minutes. Since this is quite critical in terms of model performance, an internal sub time step optimizer was implemented. The optimal sub time step for heat transfer may be shorter or longer than the optimal time step for the Richards approach (which is estimated by evaluating the Courant-condition). The minimum of both sub time steps will be the winner, so the model will satisfy the heat transfer as well as the water transport. The following algorithm is applied to each soil layer, to each cell, and in each time step to get the optimal sub time step:

    1. running the complete scheme down to point 6 (see above description of the numeric scheme)

2. computing the so called mesh Fourier number F_M which must be less than 0.5 in order to prevent the solution from oscillating and getting the maximum time step by:

$$F_M = \frac{\lambda \cdot \Delta t}{(\Delta x)^2} \quad \rightarrow \quad with\, F_M \leq 0.5 \quad \rightarrow \quad \Delta t \leq \frac{0.5 \cdot (\Delta x)^2}{\lambda} \qquad \textbf{(2.24.15)}$$

3. The split factor (as quotient of the nominal time step and the Δt from the above equation) can be limited by model parameters: the minimum sub time step and the maximum sub time step can be set manually. So the computed optimal split factor can be limited in both directions in order to prevent numeric instabilities. It is recommended to make some tests using the test data set and playing with the soil parameters and other input (rain etc.). The recommended values are between 3 seconds for the minimum sub time step and around 60 to 180 second for the maximum sub time step for the explicit solution, depending on layer thickness and hydraulic parameters. When using the implicit solution, the minimum sub time step may also be much longer, since the approach is more stable (up to 1800 seconds or even more → make tests!).
4. This split-factor is then compared with the split-factor computed for the Richards approach. The larger value will be taken as effective split factor for the actual cell.

2) Limiting the heat transfer: If for some reason the sub time step was not sufficient to keep the model on safe ground (which means: no numerical instabilities), another backup algorithm is taking over. This may happen because the minimum or maximum allowed sub time step was set wrongly or because the gradients are beyond the assumptions (only top to bottom gradients were checked). There are four checks executed for each numeric soil layer:

1. if the heat is flowing upwards through the layer (see equations 2.24.12 and 2.24.13 and figure Fehler: Verweis nicht gefunden for reference):

$$\text{if } (Q_{in} \geq 0 \wedge Q_{out} \geq 0) \text{ then } T_i \leq T_{i+1} \rightarrow \text{limit } T_i \text{ to } T_{i+1} \qquad \textbf{(2.24.16)}$$

if the heat if flowing out to the upper as well as to the lower layer:

$$\text{if } (Q_{in} \geq 0 \wedge Q_{out} < 0) \text{ then } T_i \geq min(T_{i+1}, T_{i-1}) \rightarrow \text{limit } T_i \text{ to } T_{i+1} \text{ or to } T_{i-1} \qquad \textbf{(2.24.17)}$$

2. if the heat is flowing upwards through the layer

$$\text{if } (Q_{in} < 0 \wedge Q_{out} < 0) \text{ then } T_i \leq T_{i-1} \rightarrow \text{limit } T_i \text{ to } T_{i-1} \qquad \textbf{(2.24.18)}$$

3. If the heat is flowing into the layer from the upper layer as well as from the lower layer:

$$\text{if } (Q_{in} < 0 \wedge Q_{out} \geq 0) \text{ then } T_i \leq max(T_{i+1}, T_{i-1}) \rightarrow \text{limit } T_i \text{ to } T_{i+1} \text{ or to } T_{i-1} \qquad \textbf{(2.24.19)}$$

However, this mechanism cannot prevent the model completely from numeric instabilities, it can only prevent the model from "exploding". A better way to prevent numerical instabilities is to use the implicit time-stepping solution (backward Euler integration), which is briefly described in the following paragraphs.

2.25.7.3 Solution schemes

There are two possible solution schemes: the explicit and the implicit solution scheme.
The **explicit scheme** is also known as explicit time stepping or the forward Euler integration. It is straightforward and uses exactly the equations given above (but without advection!). The temperatures at time n+1 are depending solely on the temperatures at time n, so the solution can be directly taken from the solution of equation (2.24.14) .

However, the explicit solution tends to be numerically unstable, especially if limiting the sub time

step length to a given minimum. This means that a model with an explicit solution will need more time to run in a stable state. On the other hand, an **implicit solution** (also known as backward Euler integration or implicit time stepping) uses the (not known) temperatures at time n+1 to define the temperatures at any layer. This results in a linear system of equations which is shown here as an example:

Let's use eq. (2.24.11) with some modifications:

$$\Delta T_i = T_{i,n+1} - T_{i,n} = \frac{\Delta t}{x_i \cdot C_i} \cdot \left(Q_i - Q_{i-1}\right) - q_{i-1} \cdot c_w \left(T_{i,n} - T_{i-1,n+1}\right) - q_{i+1} \cdot c_w \left(T_{i+1,n+1} - T_{i,n}\right) - q_{mac} \cdot c_w \left(T_{mac} - T_{i,n}\right) \quad \textbf{(2.24.20)}$$

where
- T is the temperature in layer i [°C]
- i denotes the numerical layer index
- n denotes the time index (n is actual time, n+1 is one sub time step later)
- Δt is the sub time step (see eq. (15)) [s]
- x is the layer thickness [m]
- C is the actual heat capacity of that layer [J/(m^{3}K)]
- Q is the actual heat flux between the layers (index i denotes flux between layer i+1 and i, index i-1 denotes flux between layer i-1 and I)
- q_{i-1} is the water flux between the upper and the actual layer; upward is positive [m/s]
- q_{i+1} is the water flux between the lower and the actual layer; upward is positive [m/s]
- q_{mac} is the water flux from macropore infiltration (always >= 0) [m/s]
- c_w is the specific heat capacity of liquid water [J/(m^{3}K)]
- T_{mac} is the temperature of macropore infiltration water, which is set to either 0°C from snow melt or to the air temperature for infiltrating rain water [°C]

By using the individual thicknesses and thermal conductivities as well as actual heat capacities of each layer (remember: each numerical layer can have a distinct set of parameters), eq. (2.24.20) can be written as:

$$T_{i,n+1} - T_{i,n} = \frac{\Delta t}{x_i \cdot C_i} \cdot \left[\frac{\lambda_i}{-0.5 \cdot \left(x_i + x_{i+1}\right)} \cdot \left(T_{i,n+1} - T_{i+1,n+1}\right) - \frac{\lambda_{i-1}}{-0.5 \cdot \left(x_i + x_{i-1}\right)} \cdot \left(T_{i-1,n+1} - T_{i,n+1}\right) \right] - q_{i-1} \cdot c_w \left(T_{i,n} - T_{i-1,n+1}\right) - q_{i+1} \cdot c_w \left(T_{i+1,n+1} - T_{i,n}\right) - q_{mac} \cdot c_w \left(T_{mac} - T_{i,n}\right)$$

which, after isolating all factors for $T_{i-1,n+1}$, $T_{i,n+1}$ and $T_{i+1,n+1}$, resp., can be written as:

$$a_i \cdot T_{i-1,n+1} + b_i \cdot T_{i,n+1} + c_i \cdot T_{i+1,n+1} = T_{i,n} - \frac{\Delta t}{x_i \cdot C_i} \cdot q_{mac,i} \cdot c_w \cdot T_{mac} \quad \textbf{(2.24.21)}$$

with

$$a_i = -\frac{2 \cdot \Delta t \cdot \lambda_{i-1}}{x_i \left(x_i + x_{i-1}\right) \cdot C_i} - \frac{q_{i-1} \cdot c_w \cdot \Delta t}{x_i \cdot C_i}$$

$$c_i = -\frac{2 \cdot \Delta t \cdot \lambda_i}{x_i \left(x_i + x_{i+1}\right) \cdot C_i} + \frac{q_{i+1} \cdot c_w \cdot \Delta t}{x_i \cdot C_i} \quad \textbf{(2.24.22)}$$

$$b_i = 1 - a_i - c_i - \frac{q_{mac} \cdot c_w \cdot \Delta t}{x_i \cdot C_i}$$

Such a linear equation as given in eq. (2.24.21) can be formulated for each of the m numerical layers, resulting in m equations with m+2 unknowns. Since the upper and the lower boundary

conditions as well as all terms on the right side of (2.24.21) are known, the number of unknowns is reduced to *m*, thus resulting in a set of *m* linear equations with *m* unknowns ($T_{i-1,n+1}$ for layer 1 is the known upper boundary condition, $T_{i+1,n+1}$ for the bottom layer is the lower boundary condition, the respective terms with their factors are moved to the right hand side). All the other known right hand values are the temperatures at time *n*. This set of equations can now be seen as a tri-diagonal or tri-banded matrix which has to be solved to get the temperatures at time *n*+1. The solution of such a tri-banded matrix is computationally very effective (and is not described here, see e.g. Wikipedia for reference). Note that the equations (2.24.21) and (2.24.22) will turn into the pure heat conduction form when there is no advection, since all the additional terms will then be zero.

Note: Equations (2.24.21) and (2.24.22) look a bit different for the uppermost and lowermost layer, since

- the temperatures $T_{i-1,n+1}$ for the top layer and $T_{i+1,n+1}$ for the bottom layer are the known boundary conditions
- the distance between the center of the top and bottom layer to the respective boundary is only one half of the respective layers thicknesses
- factors a (for upper boundary condition) and c (for lower boundary conditions) are set to zero (in the tri-banded matrix' left hand side, not on the right hand side) in order to have a reasonable systems of equations.

Thus, the top layers equation looks like this:

$$
\begin{aligned}
b_1 \cdot T_{1,n+1} + c_1 \cdot T_{2,n+1} &= T_{1,n} - \frac{\Delta t}{x_1 \cdot C_1} \cdot q_{mac,1} \cdot c_w \cdot T_{mac} - a_1 \cdot T_{0,n+1} \\
&\text{with} \quad x_0 = 0 \text{ and } T_{0,n+1} = T_{air} \text{ for no snow and 0°C for snow:} \\
b_1 \cdot T_{1,n+1} + c_1 \cdot T_{2,n+1} &= T_{1,n} - \frac{\Delta t}{x_1 \cdot C_1} \cdot q_{mac,1} \cdot c_w \cdot T_{mac} + T_{air} \cdot n_f \cdot \left[\frac{2\Delta t \cdot \lambda_1}{x_1^2 \cdot C_1} + \frac{q_0 \cdot c_w \cdot \Delta t}{x_1 \cdot C_1} \right]
\end{aligned}
\qquad \textbf{(2.24.23)}
$$

where q_0 in the braced term is the infiltrating water and $T_{0,n+1}$ was replaced by applying the air temperatur T_{air} and the f-factor n_f.

Accordingly, the equation for the bottom layer can be written like this:

$$
\begin{aligned}
a_i \cdot T_{i,n+1} + b_i \cdot T_{i,n+1} &= T_{i,n} - \frac{\Delta t}{x_i \cdot C_i} \cdot q_{mac,i} \cdot c_w \cdot T_{mac} + T_{low} \cdot \left[\frac{2\Delta t \cdot \lambda_i}{x_i \cdot C_i \cdot (x_i + x_{i+1})} - \frac{q_{i+1} \cdot c_w \cdot \Delta t}{x_i \cdot C_i} \right] \\
&\text{with } x_{i+1} = 0 \text{ and } q_{i+1} = 0: \\
a_i \cdot T_{i,n+1} + b_i \cdot T_{i,n+1} &= T_{i,n} - \frac{\Delta t}{x_i \cdot C_i} \cdot q_{mac,i} \cdot c_w \cdot T_{mac} + T_{low} \cdot \frac{2\Delta t \cdot \lambda_i}{x_i^2 \cdot C_i}
\end{aligned}
\qquad \textbf{(2.24.24)}
$$

If the lower boundary condition is a constant heat flux, which may not necessarily be constant for the entire model run but temporally variable by reading in a new standard grid every month or so for the " `HeatFlux_Lower_Boundary_Condition` "-grid), equation (2.24.24) can still be used, although the lower boundary condition is used as constant temperature there. The solution is simple: A constant heat flux can be defined like this:

$$
q_{low} = \lambda_i \cdot \frac{\Delta T}{x_i \cdot 0.5} \qquad \textbf{(2.24.25)}
$$

where q_{low} is the heat flux through the lower boundary into the bottom layer of the heat transfer and soil model in W/m² or J/(s·m²); λ_i is the thermal conductivity of the bottom layer in W/(m·K) or

J/(s·m·K); x_i is the layer thickness of the bottom layer (and 0.5 is a factor to account for the fact that the flux is calculated between the lower boundary and the center of the bottom layer only). To get an estimation for the lower boundary temperature that creates this flux, the term ΔT is split into $T_{low} - T_i$ and then T_{low} is isolated:

$$T_{low} = T_{i,n} + \frac{x_i \cdot q_{low}}{2 \cdot \lambda_i} \qquad \textbf{(2.24.26)}$$

So if a constant heat flux is used instead of a constant temperature method, equation (2.24.26) replaces T_{low} in equation (2.24.24) before solving the equation system.

The big advantage of such an implicit solution is the numerical stability. This allows for longer sub time steps to be used. However, one should be careful to set the sub time step not to a too large value, since the specific heat capacity can be, depending on the van-Genuchten parameters, quite sensitive to temperature changes, resulting in incorrect temperatures when using too long time steps. The disadvantage of the implicit solution is the in-accurateness when the time steps get too long (which risk can of course be minimized by choosing shorter minimal sub time steps).

A series of tests resulted in the recommendation that for daily time step the sub time step for explicit solutions should not be larger than 1800 seconds, whereas for implicit solutions, the time step can still be as large as a day without any substantial loss of accuracy. For hourly resolution, the situation is similar: explicit solution should not use larger time steps than 1200 s whereas for implicit solutions 3600 s are still useful. However, some artificial effects will be seen in both versions because of the larger temporal temperature gradients within a day, so shorter time steps are recommended as down to 900s for both methods when running in an hourly time step. Shorter input data resolution as well as thin numerical layers may require even shorter sub time steps.

2.25.8 Some Examples

2.25.8.1 Basics for applying the heat transfer model

For testing the heat transfer model and investigating the effects of different parameters on soil temperatures, a test data set is available from the author. The following examples are created using this test data set.

The test “basin” is a 10 x 10 cell area, 100 m cell size, 1 to 5 m elevation. The cell at row 5 and column 5 is the control plot.

The basin was set so that at almost every time step, all soil layers are saturated: cold climate, small evaporation amounts, regular precipitation (0.25mm per day), so there is no chance for drying out. Air temperature was represented by a sine-function of +/- 15°C annual amplitude for daily values, overlaid by a sine function with +/- 10°C for hourly values (so the temperatures vary between -25°C and -5°C at the end of December and between +5°C and +25°C at the end of June for hourly values). The input air temperatures are shown in figure 2.25.8.1 and figure 2.25.8.3. The tests were done with hourly and daily resolution.

The soil has 3 horizons with a 20cm organic top layer (10 layers of 2 cm each). Below that top horizon, 10 layers of 0.05 cm each form the second horizon representing mineral soil (50 cm thick) where the hydraulic conductivity is lower and the porosity is only 0.4. Below that, the third horizon reaches down to approx. 8.8 m depth through 81 layers of 10 cm each, where the lower boundary condition is defined as -10°C. The van-Genuchten parameters are identical for all layers in order to isolate the effect of heat transfer from other effects. Only the porosity of the upper horizon is set to high values as they are typical for boreal forest and arctic tundra.

Here is a ample of the soil parameterization in the soil table:

```
horizon                          = 1      2     3
DryHeatCapacity       = 810     800    900        ; # optional parameters
DryDensity            = 450     900    1450  ; # optional parameters
DryThermalConduct     = 0.3     0.5    0.57  ; # optional parameters
Name                        = Peat   SIC   SIC          ;
ksat                             = 1e-4   1e-5  1e-7   ;
k_recession                = 0.9    0.9   0.9    ;
theta_sat                  = 0.8    0.40  0.30   ;
theta_res                  = 0.05   0.07  0.07   ;
alpha                            = 1.5    1.5   1.5    ;
Par_n                            = 2      2     2      ;
Par_tau                    = 0.5    0.5   0.5    ;
thickness                  = 0.02   0.05  0.1    ;
layers                           = 10     10    81     ;
```

The air temperature is taken as upper boundary condition with a n-factor of 1.0 (no damping or amplifying). No snow model was activated in order to demonstrate the effects of the soil heat transfer model only. A more practical example follows below.

The result of the heat transfer model is shown in figures 2.25.8.3 and 2.25.8.4. Several interesting effects can be observed:

- damping of amplitude: the deeper the soil layer, the smaller the amplitude of the temperature changes
- time-shift of amplitude: the deeper the soil layer, the later the minimum and maximum occur
- larger temperature amplitudes for frozen soils compared to thawed soils (e.g. The red line in figure 2.25.8.2)
- no diurnal fluctuations below 50 cm (thawed) and 80-90 cm (frozen) resp.
- Almost no diurnal fluctuations during phase change
- phase change takes a long time (much energy must be transported into the soil to supply the latent heat for melting. Vice versa, during freezing, a lot of energy is set free, delaying the freezing of the soil for a considerable amount of time
- the last effect in conjunction with the time shift leads to the situation that almost the entire thawed soil starts to freeze in late September / early October.
- The “wavy” shape of the temperature graphs is a numerical effect of the discretization. The temperatures are valid for the entire soil layer. So if a layer is thawing, the entire layer absorbs the latent heat for melting, thus affecting heat transfer to and from the lower and upper layer.

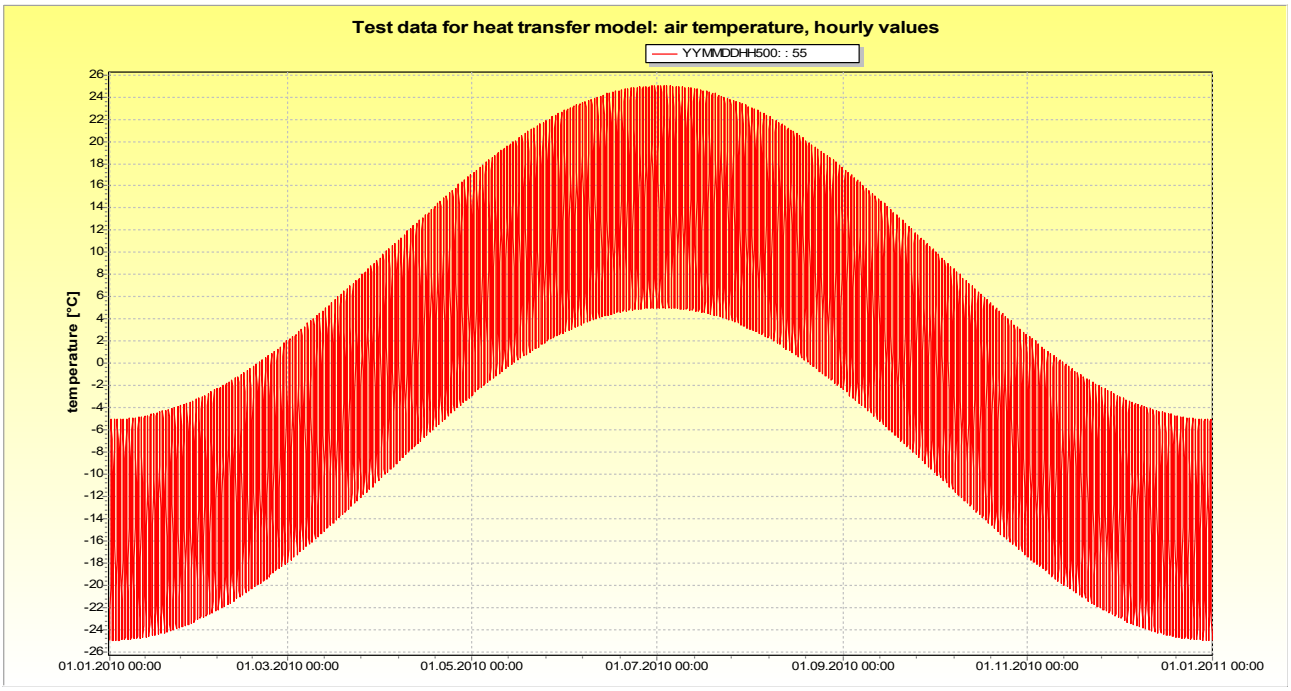


figure 2.25.8.1: temperatures used for heat transfer test data for hourly data (daily fluctuation +/- 10°C)

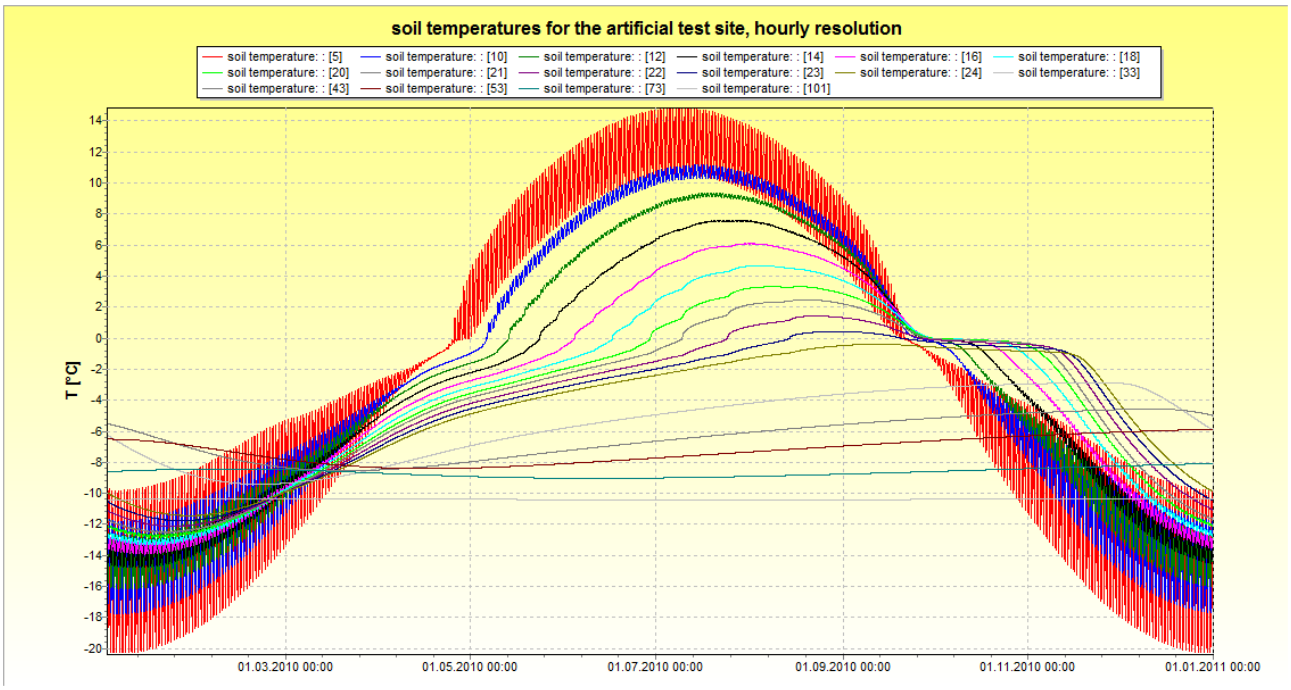


figure 2.25.8.2: soil temperatures for the artificial test site in hourly reslution (shown are temperatures for 10 cm to 110 cm in 10 cm steps and temperatures in 2, 3, 4, 7 and 8.8m)

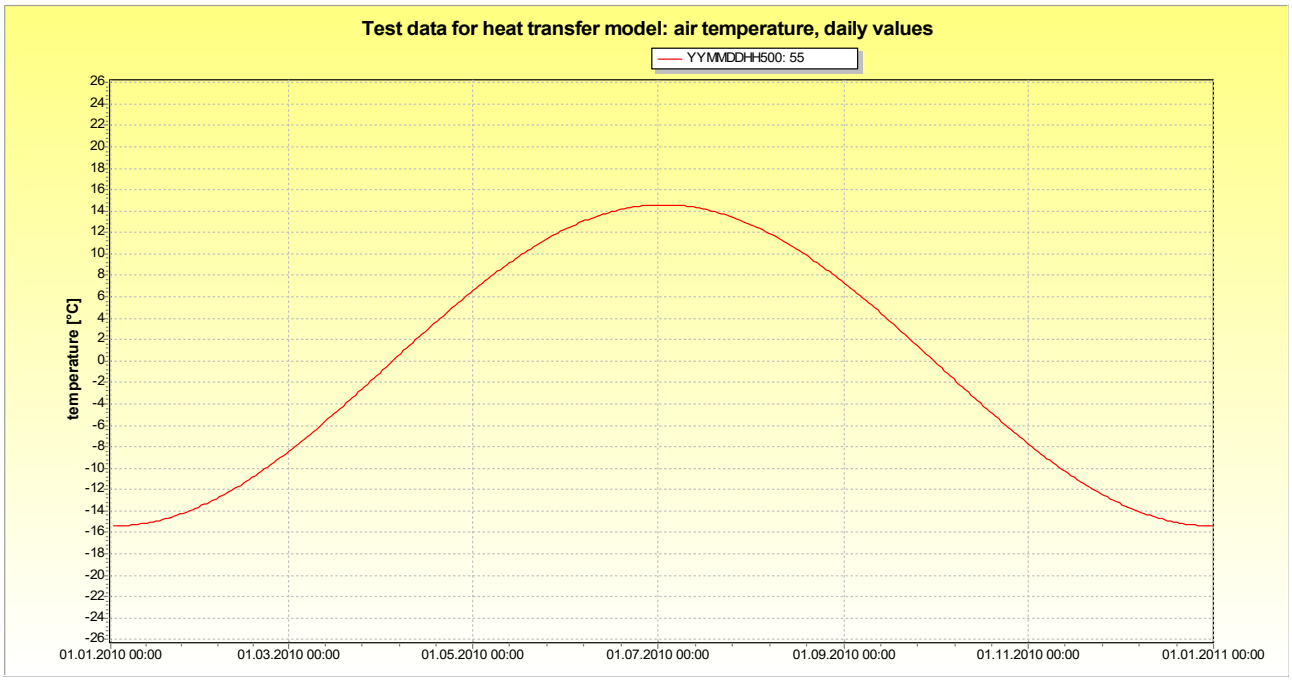


figure 2.25.8.3: temperatures used for heat transfer test data for daily data

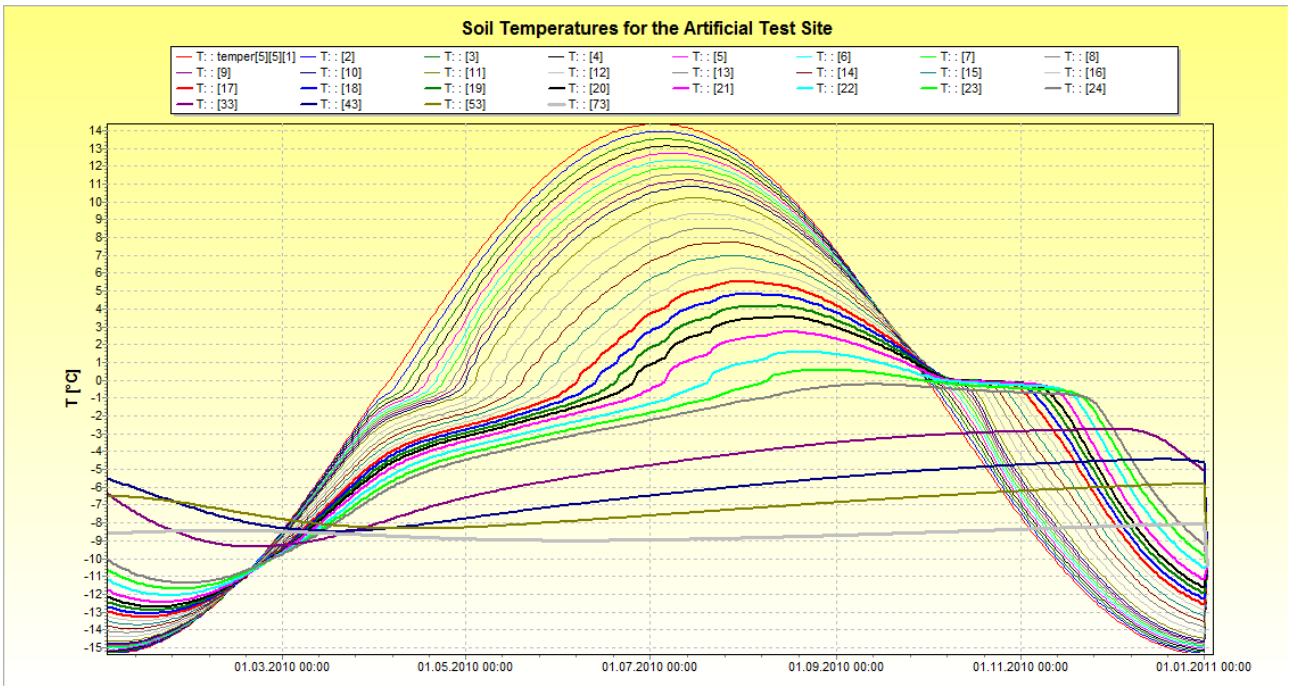


figure 2.25.8.4: soil temperatures for the artificial test site (upper horizon: 20 cm of organic material, modelled in 10 layers of 2 cm each, then 10 cm layers follow; profile has a total depth of 10 m --> the fat, dark grey graph is at around 7 m)

Figure 2.25.8.5 shows the active layer thawing and freezing as a sub-basin average. The results differ slightly: For hourly time steps, the thawing of deeper layers starts later, does consequently not reach as deep and the soils refreezes earlier. The physical background is simple: when the soil refreezes every night for a few weeks, the soil can transport more heat to the surface than it can transport from the surface into the soil at higher temperatures during the day: This is because the thermal conductivity of ice is much higher than for water. The algorithm to compute the thaw depth uses a parameter $SE_{threshold}$ which is the SE-value which should be treated as threshold for

thawing/freezing. If this parameter is set to $SE_{threshold}$=0.8, the depth where *SE* (see eq. (6)) is at most 80% (i.e. $SE \leq 0.8$) is assumed to be the thaw depth.

Note: For some soil types (like loam and clay) the thaw depth will vary considerably from the thaw depth one would expect when measuring with e.g. a metal pole, especially when setting *TSE* to lower values like 0.5. Fig. 2.25.8.5Shows the gray line as an example for hourly resolution with $SE_{threshold} = 0.8$.

Note 2: The plotting program used to create figure 2.25.8.5 (Graphlines) cannot keep track of multiple freezing fronts so it looks like the entire soil column freezes at once, which is an artifact of the plotting software. The results from WaSiM shows freezing occurring both from the ground surface and from the permafrost table in early fall.

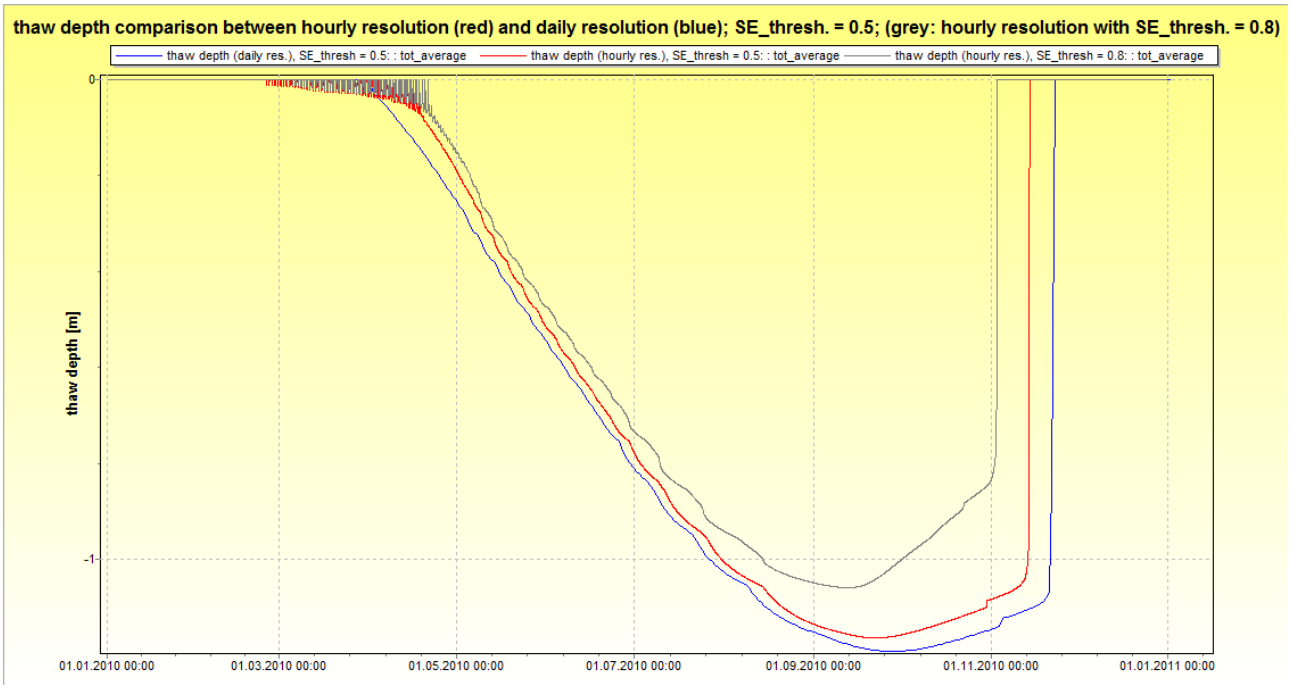


figure 2.25.8.5: Thaw depth for simulation in hourly and daily resolutions (red: hourly resolution with SEthreshold = 0.5; blue: daily resolution with SEthreshold = 0.5; grey: hourly resolution with SEthreshold = 0.8)

Another, more realistic example is shown on the following figures 2.25.8.6 and 2.25.8.7. Here, real precipitation and temperature data for Barrow, Alaska, for 2007 was used (but printed as 2010 – never mind...). After snow melt in early June (which is not met exactly, it doesn't matter as well), the generated runoff was simply averaged here for demonstration purposes.

The runoff in figure 2.25.8.6 was generated without heat transport model. The soil is completely thawed, so the interflow component is large compared to figure 2.25.8.7, where the soil was frozen for most of the time (active layer was simulated to approx. 15 cm). The model didn't use the surface routing scheme, so the effects of the shallow melt water ponds cannot be modelled here, probably the evaporation is also not modelled correctly. But that is not the point. The point is, that the frozen soil leads to completely different runoff components and also to a different distribution over time.

One comment on model performance: when using very thin soil layers like 1 or 2 cm, the sub time step must be set to very small values (down to 3 seconds). Combined with large hydraulic conductivities and high porosities for the organic layer this may lead to very long computational

times. It is therefore recommended, to use soil layers of e.g. 5 cm for the upper layer and then 10 cm or more for all other layers. If possible, the hydraulic properties of the uppermost layer should be set not to extreme values (not too high alpha and n values), in order to keep the sub time step as large as possible for the solution of the Richards equation. A run time factor of 2 to 10 with heat model compared to a model run without heat model can be expected even with such settings.

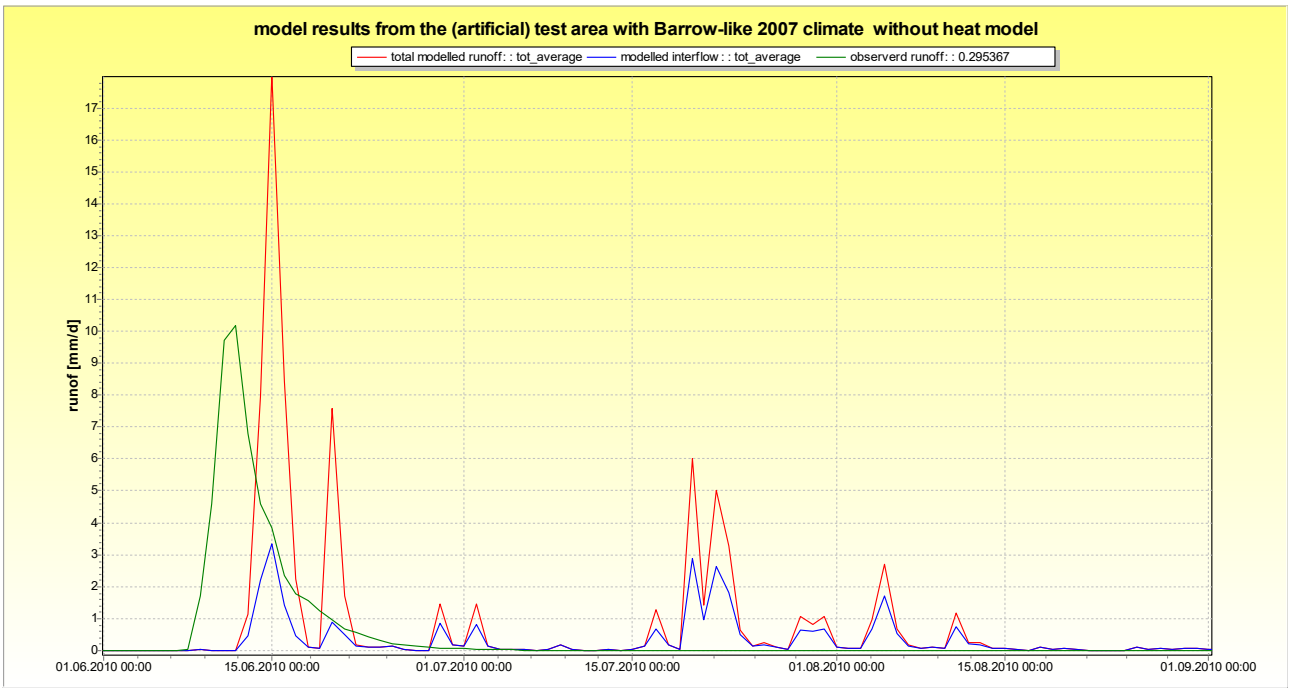


figure 2.25.8.6: Model results from an artificial arctic test site with a Barrow-like climate without the soil heat transfer model. The curves represent observed runoff (green), total modeled runoff (red), and modeled interflow (blue).

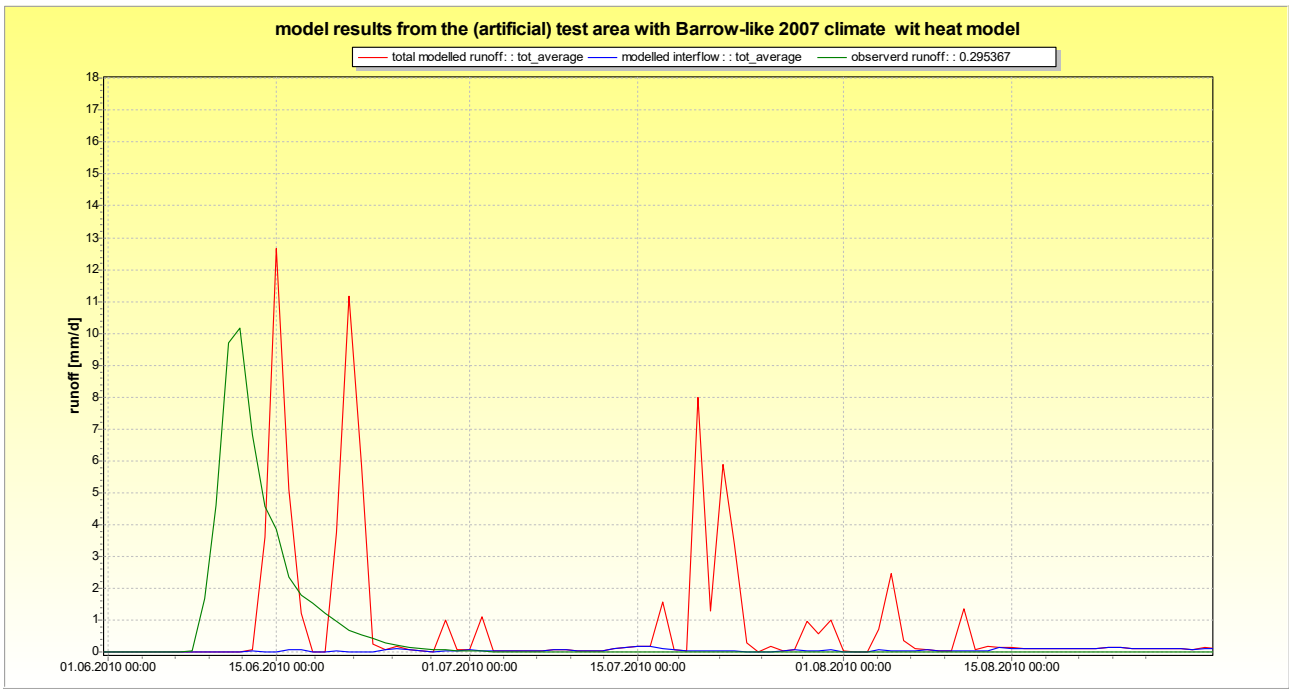


figure 2.25.8.7: Model results from an artificial arctic test site with a Barrow-like climate with the soil heat transfer model. The curves represent observed runoff (green), total modeled runoff (red) and modeled interflow(blue).

2.25.8.2 Heat transfer example for different lower boundary conditions

Figure 2.25.8.8 shows the input to the next numeric experiment. It should show the difference between fixed temperatures and a constant heat flux as lower boundary conditions.

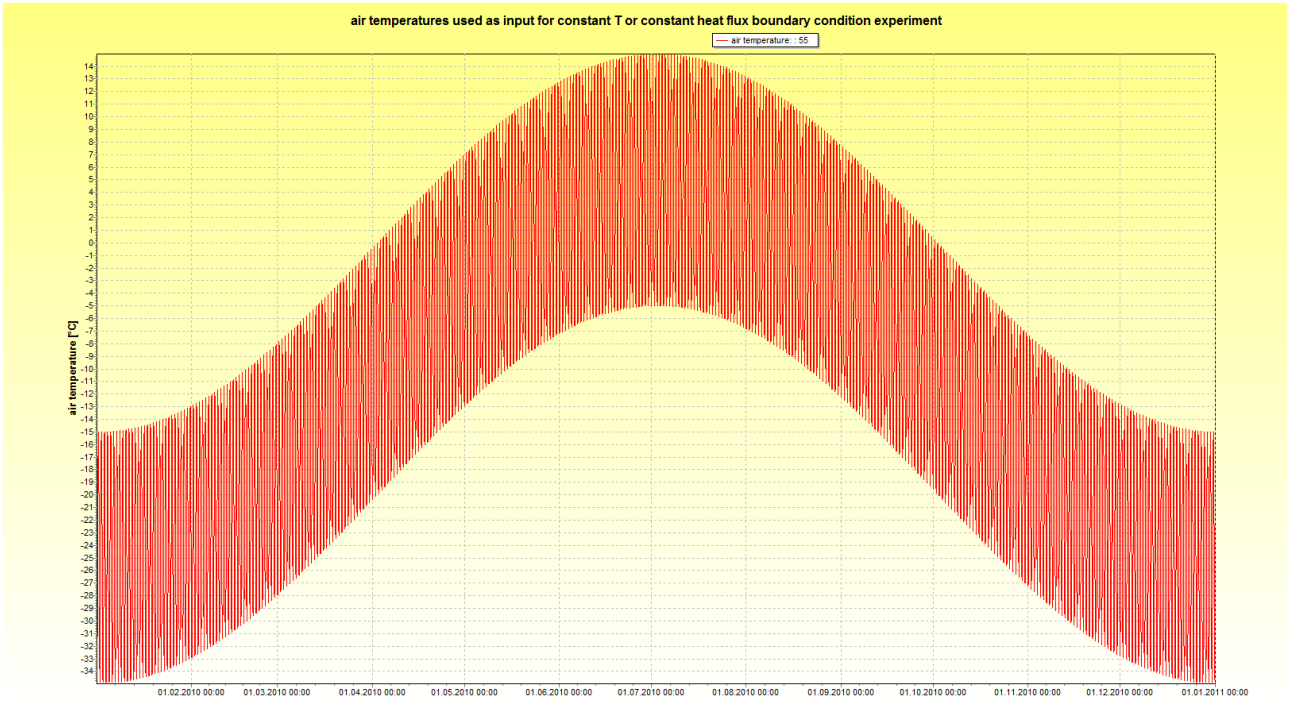


figure 2.25.8.8: air temperature for the following experiments. T ranges from -25°C in winter to +5°C in summer with diurnal fluctuations of +/- 10°C. Annual average is -10°C

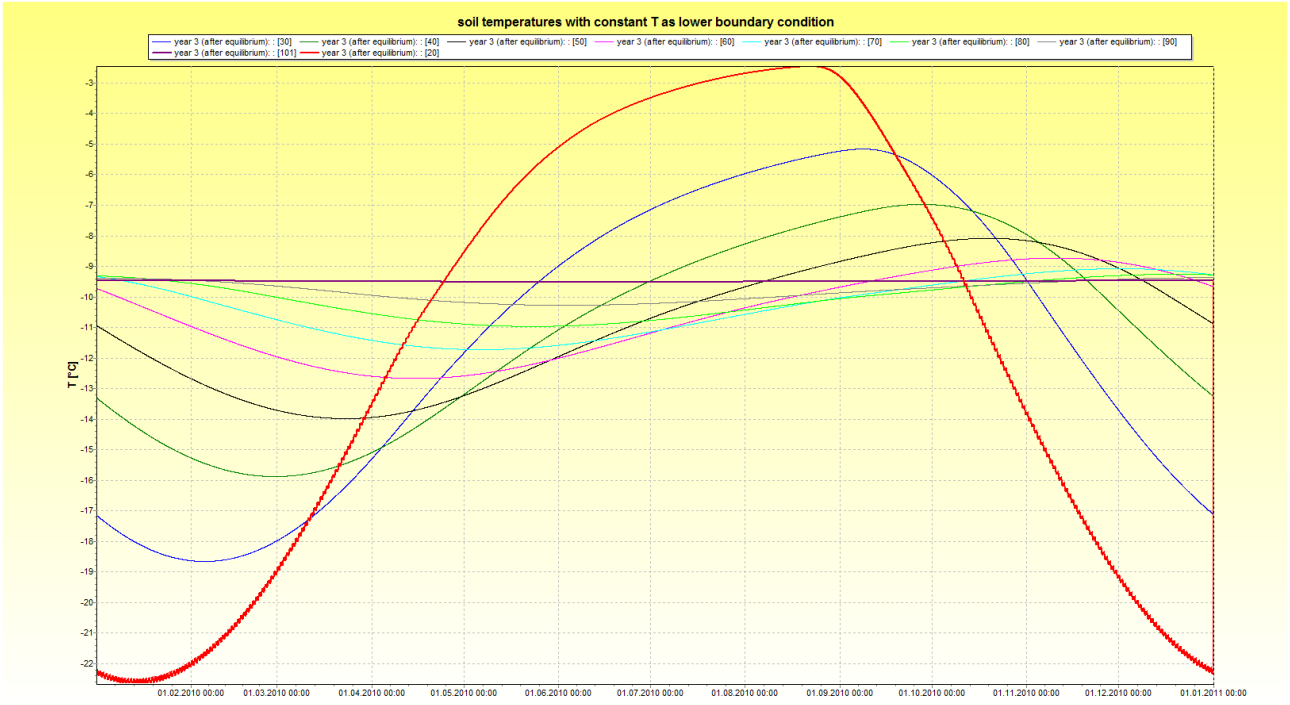


figure 2.25.8.9: soil temperatures as time series for constant T lower boundary condition (red=0.7m, blue = 1.7m, green = 2.7m, black = 3.7m etc. down to 8.8m for violet)

As can be seen in figure 2.25.8.9 the lower boundary has a constant temperature of approx. -9.5°C.

When changing the lower boundary condition to a constant heat flux, that pattern changes, as can be seen in figure 2.25.8.10. Due to the deep temperatures of the entire soil, the temperatures at the lower boundary are even decreasing, despite the constant heat flux of 65 mW/m^2.

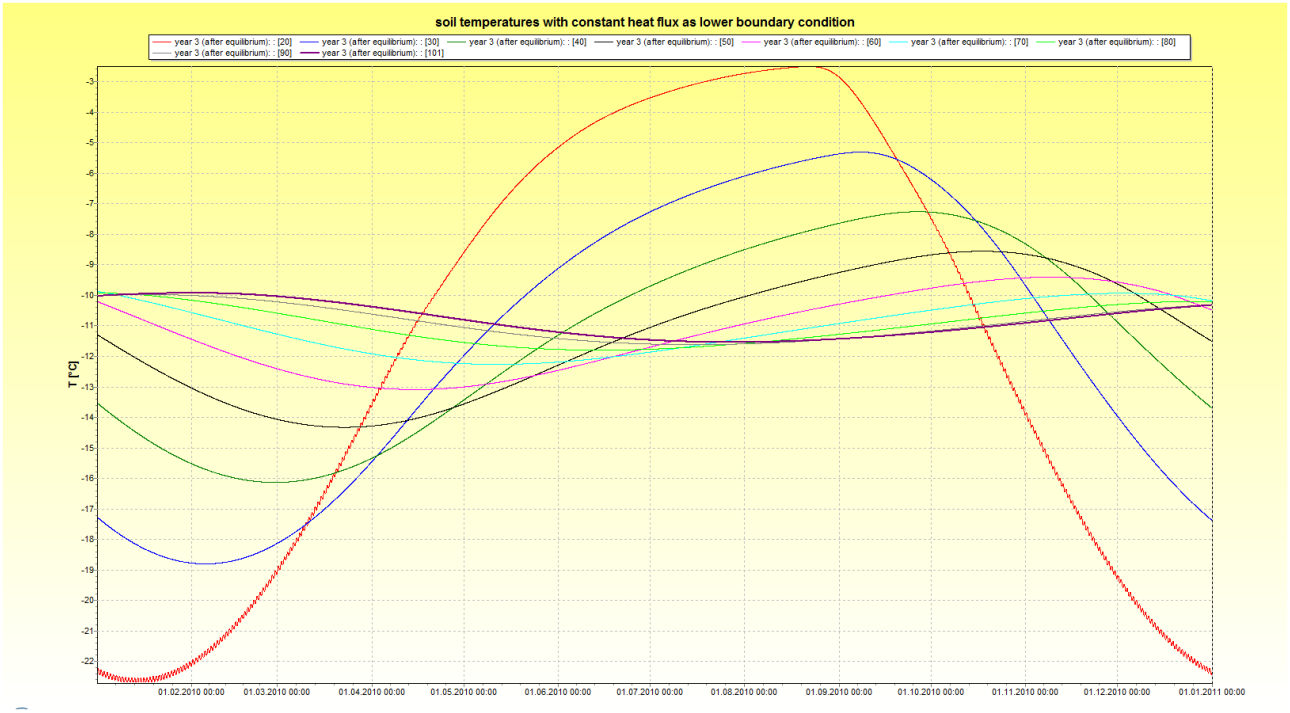


figure 2.25.8.10: soil temperatures as time series for constant heat flux of 65mW/m^2 lower boundary condition (red=0.7m, blue = 1.7m, green = 2.7m, black = 3.7m etc. down to 8.8m for violet)

Another interesting comparison is shown in figures 2.25.8.11 and 2.25.8.12. Figure 2.25.8.11 shows the temperature profiles for four days in the hypothetical year 2010: January 1st, April 2nd, July 4th, and October 3rd (the choice of the four days has nothing to do with the national holidays of Slovakia (1/1), Iran (4/1), USA (7/4) and Germany (10/3) but with the minimum, maximum and inflection points of the temperature input ;-). For each of these days, the profiles for hours 1 and 13 are shown. The heat waves traveling through the soil can be clearly recognized: down to 50cm there is the diurnal variation due to the day/night changes of the surface temperature. Below that, the seasonal changes are visible. There is no significant change of temperature below ~9m. The October and April profiles are meeting at the uppermost layer – but they differ considerably in deeper layers, without any symmetry. At around -7m, the maximum temperature can be observed in winter and the minimum temperature in summer.

Figure 2.25.8.12 shows the same scenario with a constant heat flux of 65mW/m^2 as lower boundary condition. The overall temperatures are a little cooler, since 65mW is not enough to compensate for the heat loss in winter (which is bigger than the heat gain in summer due to the higher heat conductivity of ice compared to water – in this experiment, no snow is taken into account and the n-factors are always 1.0).

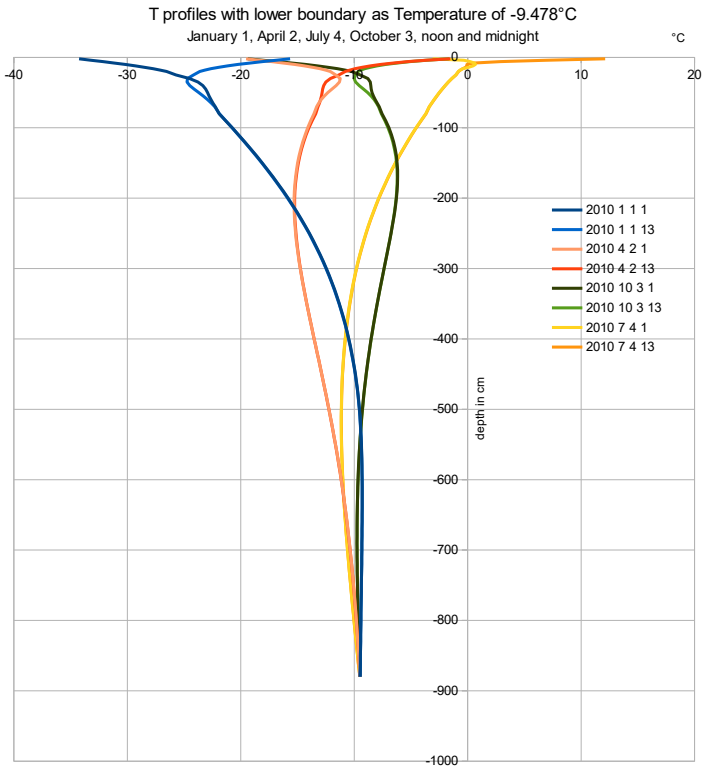


figure 2.25.8.11: T-Profile with constant T -9.478°C as lower boundary condition

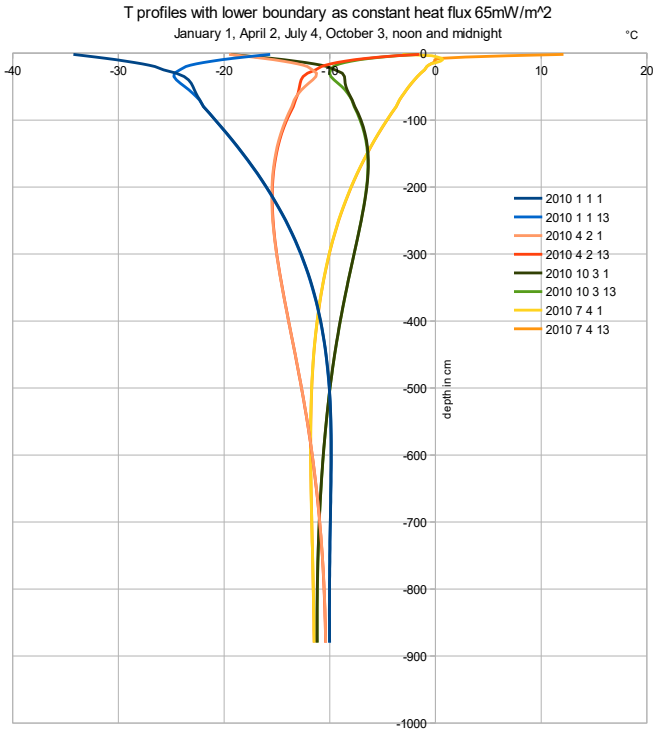


figure 2.25.8.12: T-profile with initial T of -9.478°C and constant heat flux of 65mW/m^2 as lower boundary condition

2.25.8.3 Heat transfer example for unsaturated soils

If a freezing soil is not fully water saturated, the amount of energy set free by freezing the contained water is much smaller and vice versa, when thawing the soil, the required energy to thaw the soil is much smaller than for a saturated soil. This has some consequences on the duration of the freezing/ thawing process and hence on the temperatures in the upper soil layers. Since an unsaturated soil also freezes at deeper temperatures (see equation 2.24.10)) the temperature profiles show some differences compared to saturated frozen soils.

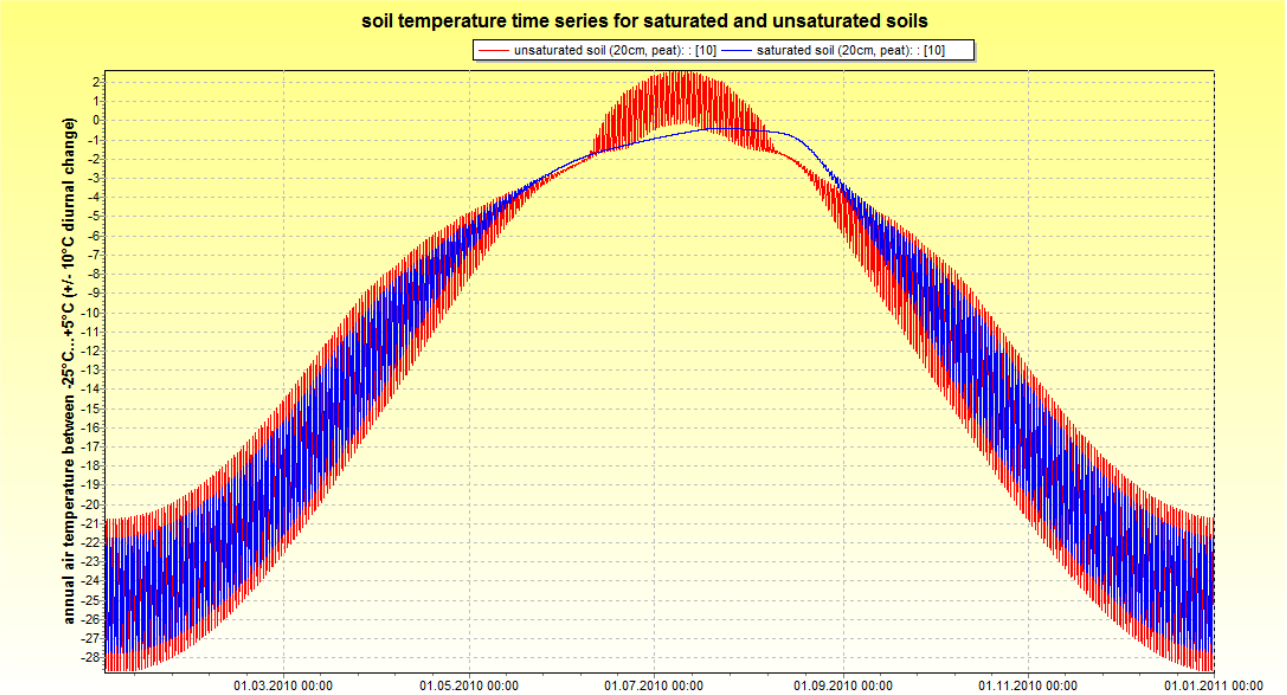


figure 2.25.8.13: Comparison of soil temperatures in 20 cm depth for a saturated (blue) and unsaturated (red) peat (porosity 0.8);

Figure 2.25.8.13 shows a comparison between a saturated (blue) and an unsaturated (red) soil, each in 20cm depth, simulated with identical temperature input (see figure 2.25.8.1). The soil profile is a silty clay with a 20cm peat layer on top. Note that for unsaturated soil the melting temperature of the soil water is around -1.8°C (below and above, the large fluctuations show that there is no more ice to thaw) whereas for the saturated soil the melting temperature is sharply below 0°C (no fluctuations to bee seen because there is still ice to be molten). The generally larger diurnal fluctuations of the temperatures in the unsaturated soil are because of the lower total heat capacity (lower water content).

Figures 2.25.8.14 and 2.25.8.15 show the respective temperature curves in 45 cm and 100 cm depth. Because of the larger water content, the thermal inertia of the saturated soil is much higher – which is manifested by the lesser temperature reached in summer and the slower cool down in fall.

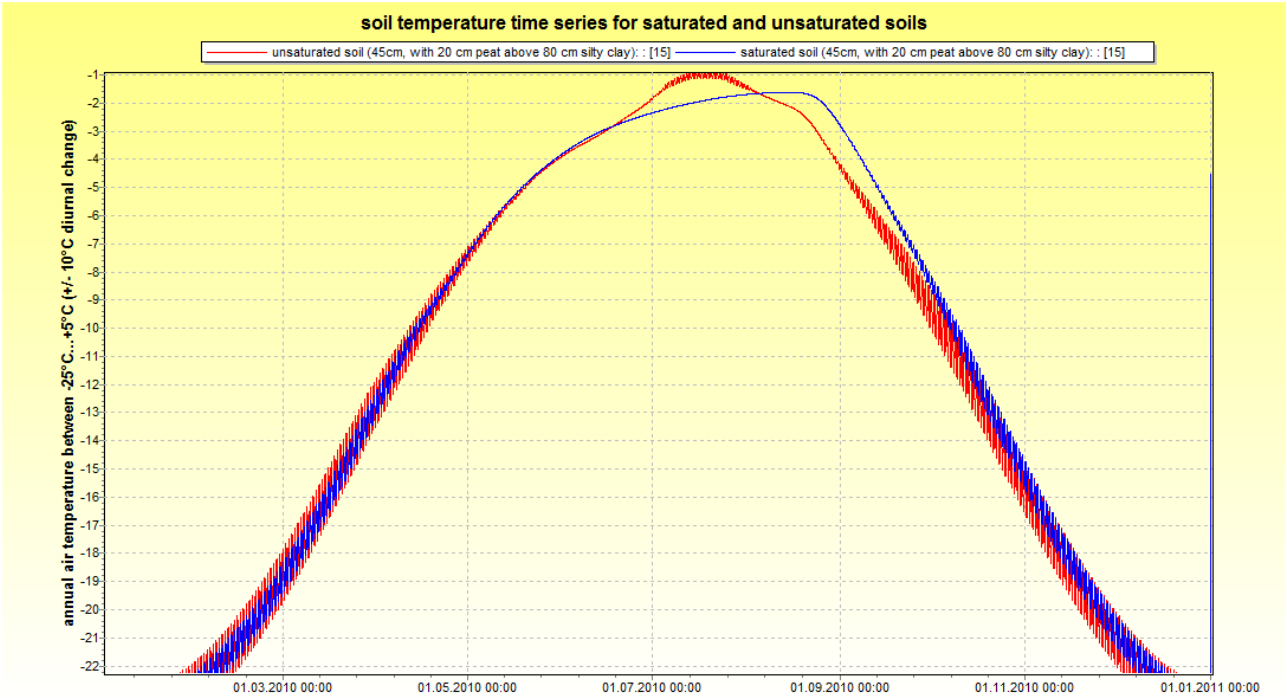


figure 2.25.8.14: Comparison of soil temperatures in 45 cm depth for a saturated (blue) and unsaturated (red) soil with 20 cm peat (porosity 0.8) above 25 cm silty clay

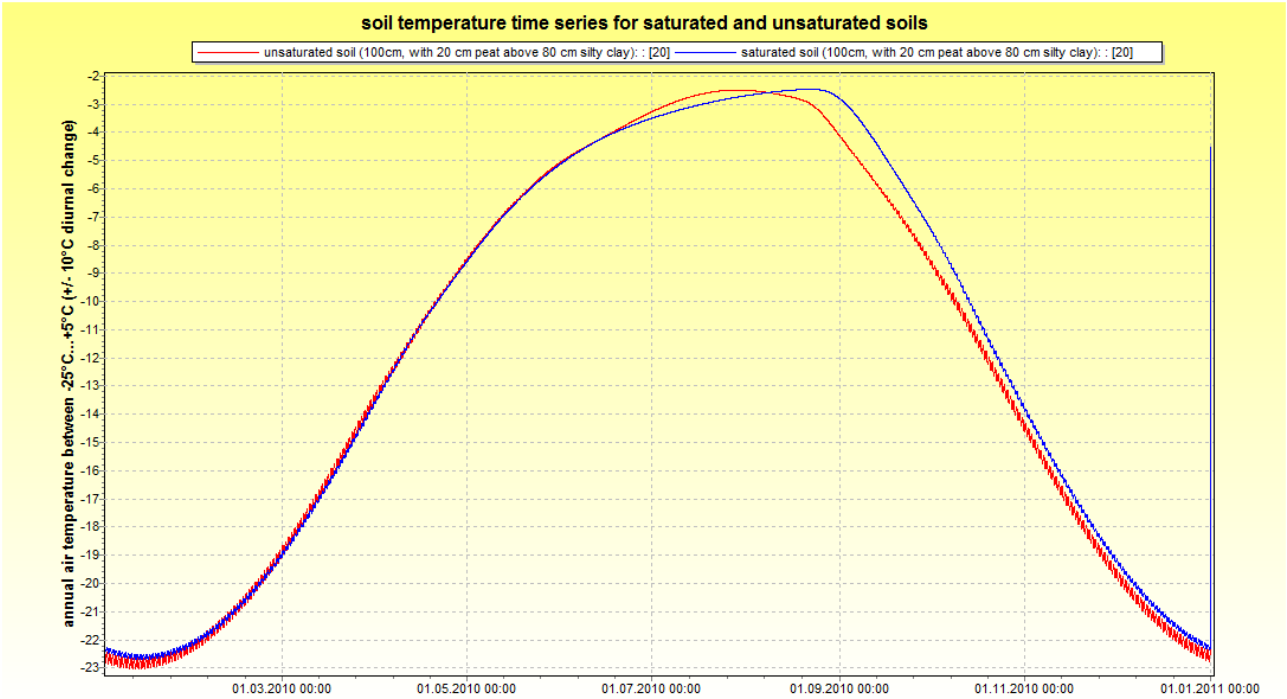


figure 2.25.8.15: Comparison of soil temperatures in 100 cm depth for a saturated (blue) and unsaturated (red) soil with 20 cm peat (porosity 0.8) above 80 cm silty clay

2.25.8.4 An experiment with a very dry peat vs. a saturated peat

Finally, the temperature profiles for saturated and unsaturated soils will be compared, see figures 2.25.8.16, 2.25.8.17 and 2.25.8.18. While figure 2.25.8.16 compares the time series of soil temperatures in 1m depth for saturated and unsaturated conditions, figures 2.25.8.17 and 2.25.8.18

show the temperature profiles for the same two model runs (again for 4 days of the year, this time the first days of January, April, July and October). Here, the uppermost 1.0 m of the soil is a peat, either saturated (water content 0.8) or unsaturated (water content ~0.19, which is near residual water content). The temperature input is the same as used for the experiments above (see figure 2.25.8.1).

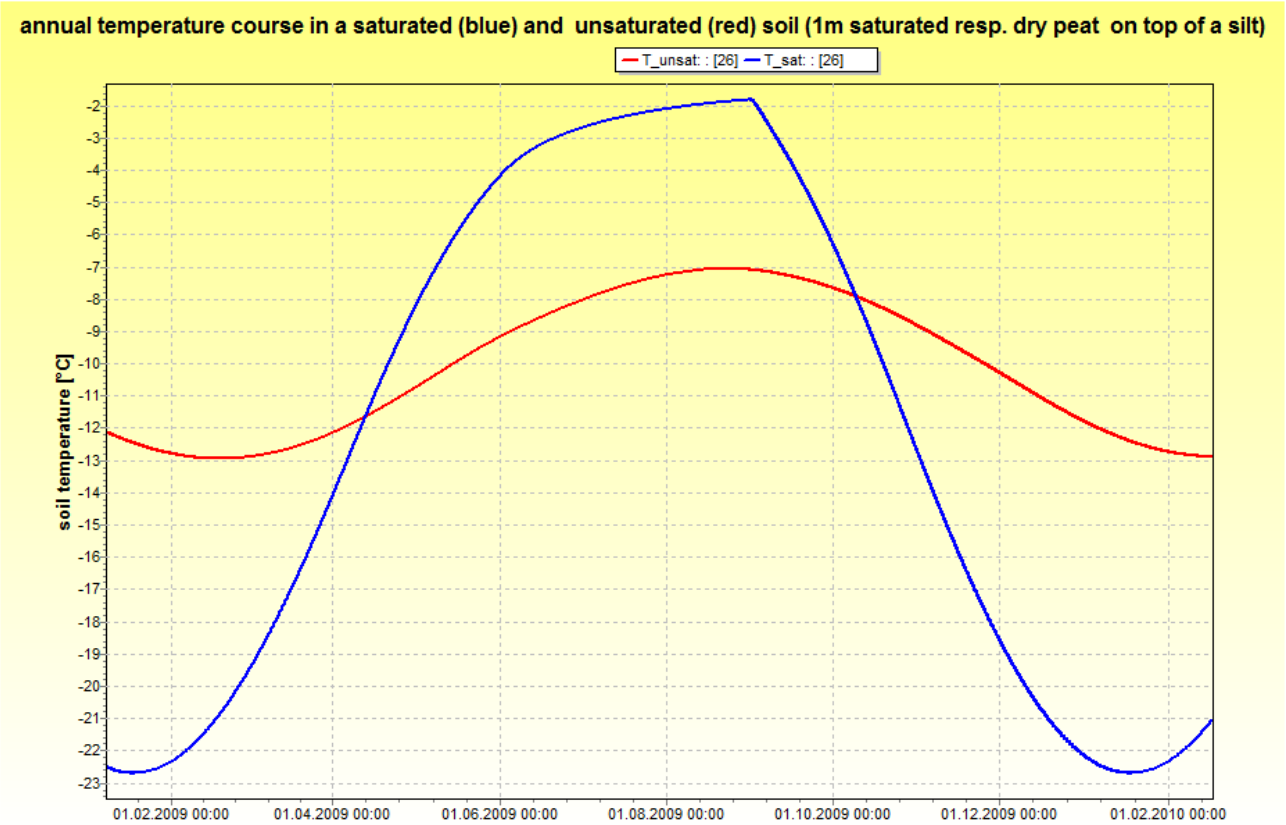


figure 2.25.8.16: temperature time series in a saturated (blue) and unsaturated (red) soil in 1m depth; the soil has a 1m peat layer on a silt. For dry peat, the groundwater is at about -1.3m. No rain, no snow, no evaporation has been regarded in this experiment, only temperature as given in the examples above

It can be seen that the unsaturated soil stays much warmer in winter and cooler in summer since the dry peat is insulating the soil from the air quite efficiently. Thus, the warming from below with 130mW/m^2 shows its effect on the unsaturated soil more than on the saturated soil: since the heat loss in winter is not as high as in the saturated case, the soil is generally warmer.

Below -10m, the thermal gradient is stable (no significant annual fluctuations). This example was calculated with a constant heat flux of 130mW/m^2, which is typical for some arctic regions in Alaska (world average is around 65mW/m^2 but the range may vary considerably)

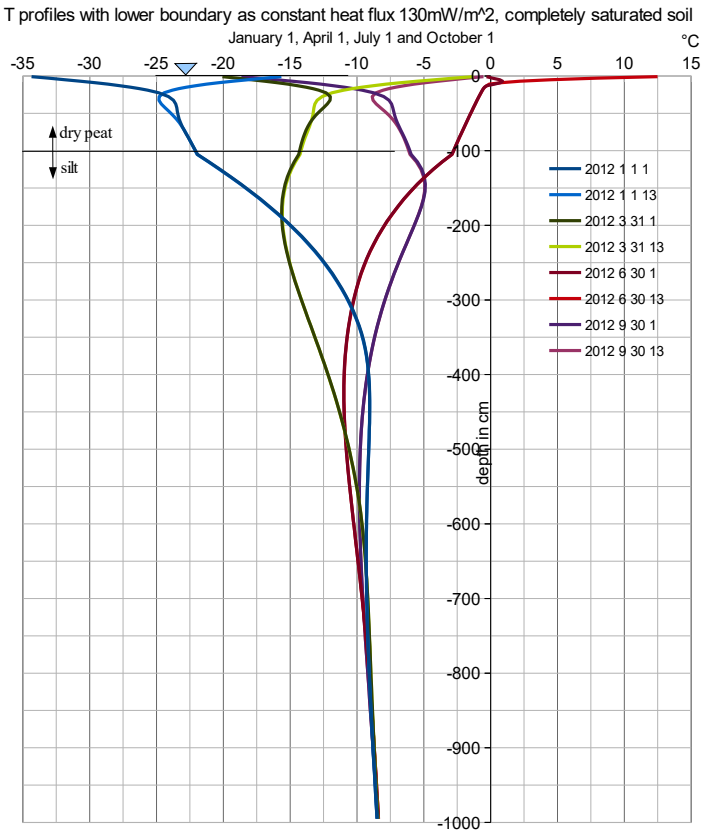


figure 2.25.8.17: T-profiles for saturated soil

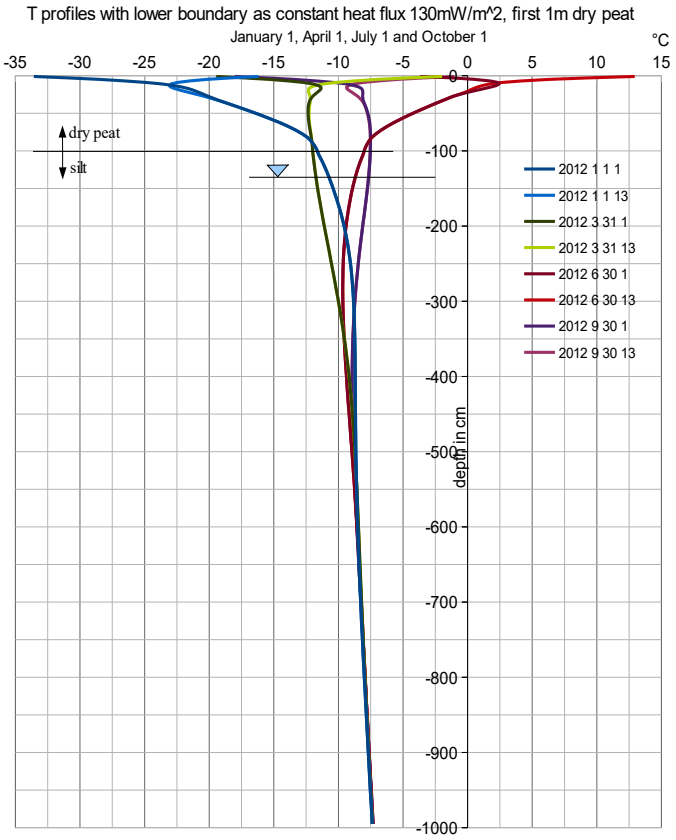


figure 2.25.8.18: T-profiles for unsaturated soil

2.25.9 Setting up the heat transfer model in the control file

There are of course a few parameters to be defined in the control file. First of all, there is a new section called [heat_transfer]. Here, all temperature modelling related parameters must be defined. In addition, some parameters can be set in the soil table to individual values per soil, overriding the global settings in the [heat_transfer] section. The following example explains the changes in the control file in detail (rows ending with ↵ have a line break for this documentation only. In real control files, the single parameters must follow each other without empty lines):

```
# set a variable where all the other variables are set
$set $T_lower_boundary_grid =  $grid//.tlowbdry

# also, a variable may be defined for the result-stack for temperatures and another
#one for the grid with the active layer thickness (thaw depth)
$set $Temperaturestack  = tsoil//stack//.//$suffix
$set $ThawDepthGridTMod = thaw//$grid//.//$suffix

# an optional standard grid for the lower boundary condition may be defined here.
# if the boundary condition should change over time (e.g. because the soil depth is not
# deep enough to suppress intra-annual fluctuations), a series of boundary condition
# grids may be supplied, using the periodicity and persistent tags:
# periodicity = 30 D 24 persistent = 1
[standard_grids]
$inpath_grid//$T_lower_boundary_grid    T_Lower_Boundary_Condition fillcode = 1
defaultValue = -10 writecode = 8 readcode = 0 # periodicity = 30 D 24 persistent = 1


# the following section for heat transfer can be used with WaSiM version 9.0 ff
[heat_transfer]
1  # 0 = do not model heat transfer, 1 = heat transfer is modelled
11 # vertical 1D heat transfer in the unsaturated zone (0=no, 1=yes using explicit time↵
```

```
      stepping, only heat diffusion,  2 = yes, heat diffusion and convection (by
       infiltrating water, not yet available)), 11 = only diffusion but using an
      implicit solution scheme,12 = diffusion and convection with implicit solution
      scheme (not yet implemented)
0  #  vertical heat transfer in snow cover (not yet available)
0  #  2D lateral heat transfer by advection (coupled to water transport) in
      groundwater (not yet available)
# parameters
# the lower boundary condition for temperature may either be defined by a grid with
the internal name _T_Lower_Boundary_Condition_ or created by using the annual
temperature and the lapse rate as defined in the next two lines
-10.0  # used when no grid "_T_Lower_Boundary_Condition_" was read in only: mean annual
 air temperature reduced to sea level to be used as lower boundary condition (e.g. 5°C)
 --> used for definition of the lower boundary condition at lower soil boundary, if no
 grid with lower boundary condition was read in
-0.007       # used when no grid "_T_Lower_Boundary_Condition_" was read in only:
temperature gradient (e.g. -0.007 K/m) for defining the lower boundary condition (used
if no grid with lower boundary condition was found)
# default soil "constants": can be changed in the soil table (using DryHeatCapacity,
DryDensity and DryThermalConduct as parameter names)
800     # default heat capacity of dry soil in J/(Kg*K), default 800 --> value may be
given in detail for each soil type in the soil table
1500    # default density of dry soil in Kg/m^3 , default 1500 --> value may be given in
detail for each soil type in the soil table
0.58    # default thermal conductivity for dry soil in J/(m*s*K) or W/(m*K):
default: 0.58 --> value may be given in detail for each soil type in the soil table
1e-11      # reduced k_sat (minimum hydraulic conductivity for fully frozen soils)
# thermodynamic constants of water and ice (not for calibration! these are constants
giving only marginal room for variations)
0.5562 # thermal conductivity of liquid water (do not change, this is a matter constant)
2.33   # thermal conductivity of ice (0°C...-20°C) do not change either
4187    # heat capacity of water in J/(Kg*K) do not change as well
1940    # heat capacity of ice at -20°C in J/(Kg*K) should also not be changed
 2090    # heat capacity of ice at 0°C in J/(Kg*K) also please do not touch the value
334000  # latent heat of freezing in J/Kg this is a constant, please do not change
1000   # density of water in Kg/m^3 rather a constant (for our reasons). Do not touch!
# other parameters (not for calibrating, but there is no clear lioterature value)
  1.22  # scaling factor (solution of the clapeyron equation, literature gives values of
1.8 up to 123, bhut this may be measure dependent. Theoretical value
is dH/T_m = 1.22 J/(Kg*K))
0.8     # SE value which must be underrun to evaluate the soil layer as frozen for the
Thawdepth-output grid and statistics
60      # minimum sub time step allowed for heat transfer model (numeric errors like
extreme temperature fluctuations are possible if the value is too large)
1200    # maximum sub time step allowed for heat transfer model (to avoid instabilities
induced by the non-linearity of the processes)
1.0     # n-factor for freezing (factor applied to the air temperature to get the
temperature at the soil surface as upper boundary condition when temperatures are
negative
1.0     # n-factor for thawing (factor applied to the air temperature to get the
temperature at the soil surface as upper boundary condition when temperatures are
positive
# output grids and statistics
$outpath//ts_loc//$grid//.//$year # results soil temperature for control point
$outpath//ts_avg//$grid//.//$year $once_per_interval # results soil temperature thaw
depth or active layer thickness as average value for subbasins
$outpath//$Temperaturestack    # stack, actual soil water content for all soil levels
$Writegrid                     # Writecode for this stack
$outpath//$ThawDepthGridTMod # output grid containing the active layer thickness
$Writegrid                     # Writecode for this stack
$readgrids                   # like in all other models: 1 = read stack for temperature,
0 = create new stack according to boundary conditions (linear interpolation between
upper and lower boundary condition)

[soil_table]
1                  # number of following entries
7 silty_clay_(SIC) {method = MultipleHorizons;
....
 DryHeatCapacity    = 810    # dry heat capacity in J/(Kg*K)
 DryDensity         = 1450 # dry density in m^3/m^3
```

```
   DryThermalConduct      = 0.57 # dry thermal conductivity in W/(m*K) (or J/(m*s*K)
   KMinFrozenSoil   = 1e-12 # minimum hydraulic conductivity in m/s when the soil is
#completely frozen (do not set to zero, since the logarithm of this value is used
#internally)
...
}

# alternatively, some parameters can be provided per horizon:
[soil_table]
1                  # number of following entries
7 silty_clay_(SIC) {method = MultipleHorizons;
....
KMinFrozenSoil      = 1e-12 # minimum hydraulic conductivity in m/s when the soil is ↵
#completely frozen (do not set to zero, since the logarithm of this value is used ↵
#internally)
horizon             = 1    2      3     ; # ID of the horizon
Name                = Peat SIC10m something ;  # short descriptions
DryHeatCapacity     = 810  800    900   ; # dry heat capacity in J/(Kg*K) per horizon
DryDensity          = 450  900    1 450 ; # dry density in m^3/m^3  per horizon
DryThermalConduct   = 0.3  0.5    0.57 ; # dry thermal conductivity in W/(m*K)  per
horizon
...
}

[landuse_table]
2
1  peat_landscape {method = VariableDayCount;
RootDistr        = 1.0;
TReduWet         = 1.0;
LimitReduWet     = 0.5;
HReduDry         = 3.5;
IntercepCap      = 0.3;
JulDays          = 15   46   74   105  135  166   196   227   258   288   319   349 ;
Albedo           = 0.2  0.2  0.2  0.2  0.2  0.2   0.2   0.2   0.2   0.2   0.2   0.2 ;
rsc              = 100  100  100  100  100  100   100   100   100   100   100   100 ;
rs_interception  = 100  100  100  100  100  100   100   100   100   100   100   100 ;
rs_evaporation   = 150  150  150  150  150  150   150   150   150   150   150   150 ;
LAI              = 1    1    1    1    1    1     1     1     1     1     1     1   ;
Z0               = 1    1    1    1    1    1     1     1     1     1     1     1   ;
VCF              = 0.5  0.5  0.5  0.5  0.5  0.5   0.5   0.5   0.5   0.5   0.5   0.5 ;
RootDepth        = 0.4  0.4  0.4  0.4  0.4  0.4   0.4   0.4   0.4   0.4   0.4   0.4 ;
AltDep           = 0    0    0    0    0    0     0     0     0     0     0     0   ;
n-factor         = 0.8  0.8  0.9  0.9  0.9  1.0   1.0   1.0   1.2   1.0   0.8   0.8 ;
# factor applied to the air temperatur to estimate the soil surface temperature as
# boundary condition for the soil temperature. Values are interpolated and not altitude
# corrected
}
```

2.25.10 Additional remarks

The hydraulic conductivity of the aquifer in the groundwater model will be adjusted for frozen layers. To do this, only the saturated soil layers are examined for their temperatures. If a layer is frozen, the transmissivity for this layer is accordingly set (hydraulic conductivity for frozen soil after equation (2.24.8c) times layer thickness). All layers' transmissivities are then integrated (transmissivities added up) over the entire saturated soil (frozen and unfrozen) and the result is divided by the total saturated (frozen and unfrozen) thickness and set as new effective hydraulic conductivity for the groundwater model.

2.26 Spin-Up for heat transfer model using HRU concepts

2.26.1 Motivation for a spin-up mode

The following descriptions of spin-up and pre-spin-up are mainly based on suggestions and discussions with Ronald Daanen, Alaska Division of Geological and Geophysical Surveys,

Fairbanks, and Anna Liljedahl, University of Alaska in Fairbanks. The funding of the spin-up and HRU implementation was supported by the Environmental Sciences Division, U.S. Department of Energy, via the University of Alaska, Fairbanks (Arctic Next Generation Ecosystem Experiments “Arctic NGEE”).

Simulations of soil temperature often focus on the near surface conditions of temperature fluctuation over diurnal or annual time scales with freezing and thawing. Deeper permafrost temperature however also respond to decade, century and millennial changes in mean annual temperatures. These slow shifts in permafrost temperatures can change the energy balance of the near surface. The mean annual temperature is considered fixed over time when shorter periods are simulated and typically assigned as fixed at a depth of the zero annual amplitude of temperature swings. However when attempting to simulate longer periods of soil temperature change the flux at the depth of zero annual amplitude needs to be considered and is typically not known at that depth. The solution to this problem is to simulate the entire depth of permafrost and assigning a geothermal heat flux much below the frozen ground. This solution to the long term problem brings an additional problem of data requirement for the initial condition. An established solution to this problem is spin-up. Typically the first 10 years of climate data are used in a loop for ten cycles to create deeper permafrost temperatures that are in equilibrium with the shallow temperatures. Due to non-linearity (temperature depended thermal conductivity and heat capacity) and thermal offset (difference in temperature at the bottom of the active layer caused by the thermal conductivity difference between thawed and frozen soil state) it is hard to develop a procedure to find a fast equilibrium solution. That's where the HRU-concept is very usefull.

2.26.2 The concept of HRUs

To run as fast as possible, the model domain is not processed cell by cell but for HRU master cells only. An HRU is a **H**ydrologic **R**esponse **U**nit, i.e. a unique combination of certain properties that are hydrologically relevant. Such properties typically include the elevation (e.g. in 100m elevation belts in order to create not too much HRUs), land use class, and soil type. Also slope or exposition classes and other relevant properties like glacierized fraction, lake depths etc. may be used to form HRUs. Each HRU class is defined by its specific ID, the fingerprint, created from a combination of the chosen properties, e.g. elevations between 100 and 200m a.s.l., a land use of pasture and a soil type of loam with a soil column depth of 10m gets the ID 1, whereas a cell with an elevation between 200 and 300m and otherwise the same properties (same soil, same land use) gets the ID 2 and so on.

After constructing the HRUs, the simulation is then done only once per HRU-ID, i.e. the first occurrence of any HRU with a specific ID is the HRU-master for all cells with the same HRU-ID and will be processed like a normal grid cell, whereas all other cells that share the same HRU-ID will not be computed in detail but will get the results of each sub-model for each internal variable from their HRU-master. Thus, the model can still be used as a complete basin model which is much faster than the full grid based version running cell by cell for each time step.

However, the drawback of this method is that no lateral processes can be modeled since the location information is irrelevant to HRUs. Also, some algorithms need to be used with care since they are quasi-two-dimensional:

- don't use interpolation regions (or, if, take the regions into account when constructing HRUs)
- shadowing in radiation correction is automatically skipped when in spin-up mode
- for all snow methods including slides: slides are automatically skipped when in spin-up mode
- use wind driven snow redistribution with care: HRU construction should consider the wind driven redistribution coefficients (or switch that feature off by using one of the snow

methods without wind driven snow redistribution to be on the safe side)
- surface routing is automatically skipped when in spin-up mode
- groundwater model is automatically skipped when in spin-up mode; however, to get realistic model results, the baseflow parameters in [unsatzon_model] should be calibrated for spin-up After spin-up, when using the groundwater model, run the model for at least one or two years in normal model setup (no spin-up mode) to allow the groundwater to swing in.
- use glaciers with care: since glacier cells may be partially glacierized only, the construction of HRUs can be difficult. If glaciers are used, the glacierized fraction grid can be used for HRUs in e.g. 0.25 steps or so.

It is important to understand the complexity of the HRU approach: if the model uses many different land use classes and soil types and possibly glaciers and lakes and encompasses a large elevation range, the number of unique HRUs can be quite large. For example, a basin with appr. 12'000 cells using all of the above criteria can have up to 4'000 unique HRUs. That means that the model will be not much faster in HRU mode than in normal mode (may be by a factor of 1.5 or so, since the other cells need to get copies of the detail results in each time step). On the other hand, if the number of properties to form HRUs and their absolute count (e.g. soil types, land use classes) is small and there are no glaciers, lakes etc, it is possible to end up with a small number of HRUs down to below 10 – resulting in a tremendous speed up. So it may be worth to simplify some parameter grids for an efficient HRU spin-up.

2.26.3 Setting up Spin-Up mode and HRU creation

Both, the spin.up mode itself and the HRU creation, are controlled by a new optional section in the control file. Here is an example:

```
[SpinUp]
1      # doSpinUp: 0=no, 1=yes
1      # doTemperaturePreSpinUp: 0=no, 1=yes
5      # repeatCnt: how many times should the model run for the defined time period
5      # numGrids : number of following entries for grids to use for HRU creation. 0
# means, that a (standard) grid with identifier HydrologicRespondUnits should be read
# in, otherwise the follwoing lines will be evaluated
elevation_model 100 lin # internal grid identifier/class width/reclassification method
soil_types   1 lin # each landuse class is used for HRU creation
landuse      1 lin # each soil type is used for HRU creation
max_ponding_storage 1 lin # each lake will have at least one specific HRU
slope_angle  2 sqrt # slope classes are created with non-equidistant steps
```

explanations:

doSpinUp	The first parameter defines the usage of the spin-up mode. 0=no, 1=yes
doTemperaturePreSpinUp	0=no, 1=yes; if heat transfer model is used it is recommended to initially set the soil temperature profile by using a fast pre-spin-up method (see below)
repeatCnt	defines how many times the model runs for the defined time period (in section [model_time]) in spin-up mode. If repeatCnt is 0 and doTemperaturePreSpinUp is 1, then only the temperature pre-spin-up is done, resulting in a first guess for the soil temperature profile. That grid stack is written out as file with the name "Soil_Temperature_Stack". After renaming it to the name of the normal stack for soil temperatures, it can be used for model runs with or without spin-up (readgrids = 1). However, since the pre-spin-up

	speed is quite fast, the typical scenario will be to start a spin-up run with a pre-spin-up for soil temperatures.
NumGrids	defines the number of following entries for defining HRUs

Each grid definition consists of three parameters:

<grid identifyer> <class width> <method>	each grid to be used for HRU creation must be defined in a single line: <grid identifyer> is the identification string as internally used by WaSiM (see last pages of the documentation) <class width> is the class width to group continous values like elevations. For land use, soil tyes etc. use 1 to create HRUs for every member of the soil table and land use table resp. <method> can be *lin* or *sqrt* . *lin* means that the values of the input grid are reclassified without prior transformation, i.e. <class width> is directly applied to the input value. s *qrt* means that the input values are transformed by applying a sqare-root-transformation and then applying the linear reclassification on the transformed values (0 values will remain 0, negative values will be converted to their absolute values). *sqrt* is recommended for e.g. slope.

After creating HRUs, two grids are written to the default output deirectory:

All_HydrologicResponseUnits	HRU codes for all cells
Master_HydrologicResponseUnits	HRU master cells with their unique codes

Examples fro HRUs

The following example for HRUs was created using the settings given in the setting-example above. The total number of cells is 11'826 (see figure 2.26.3.1). By using 5 input grids to identify unique HRUs, the number of HRUs is 2'426 (see figure 2.26.3.2).

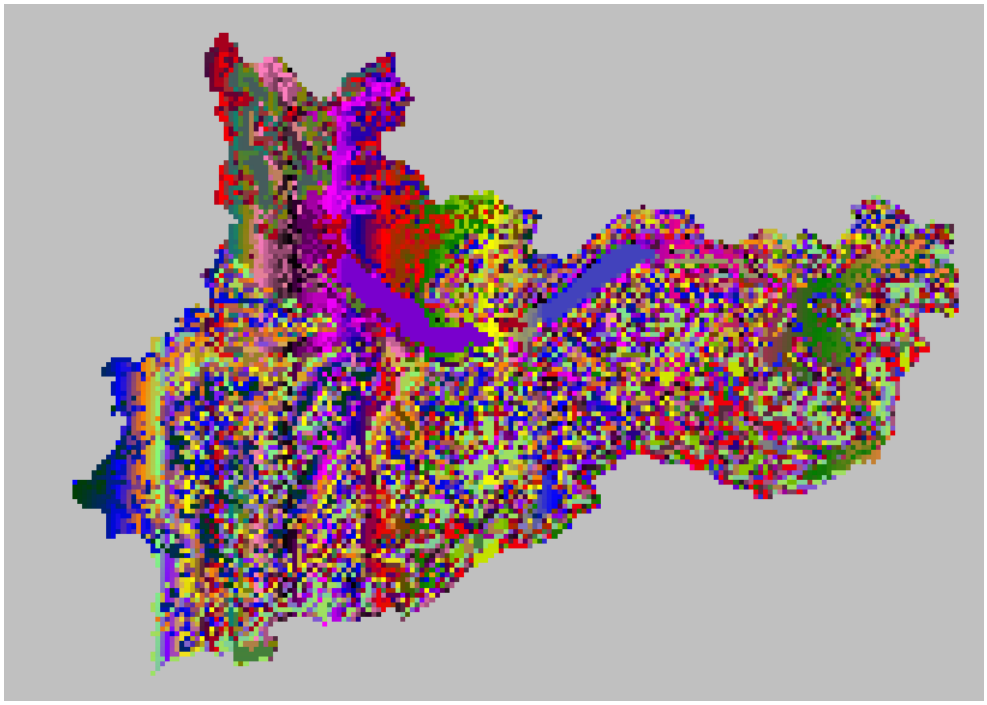


figure 2.26.3.1: HRUs for all cells (basin of the river Aare up Berne, Switzerland, 11826 cells (=2956.5 km2)

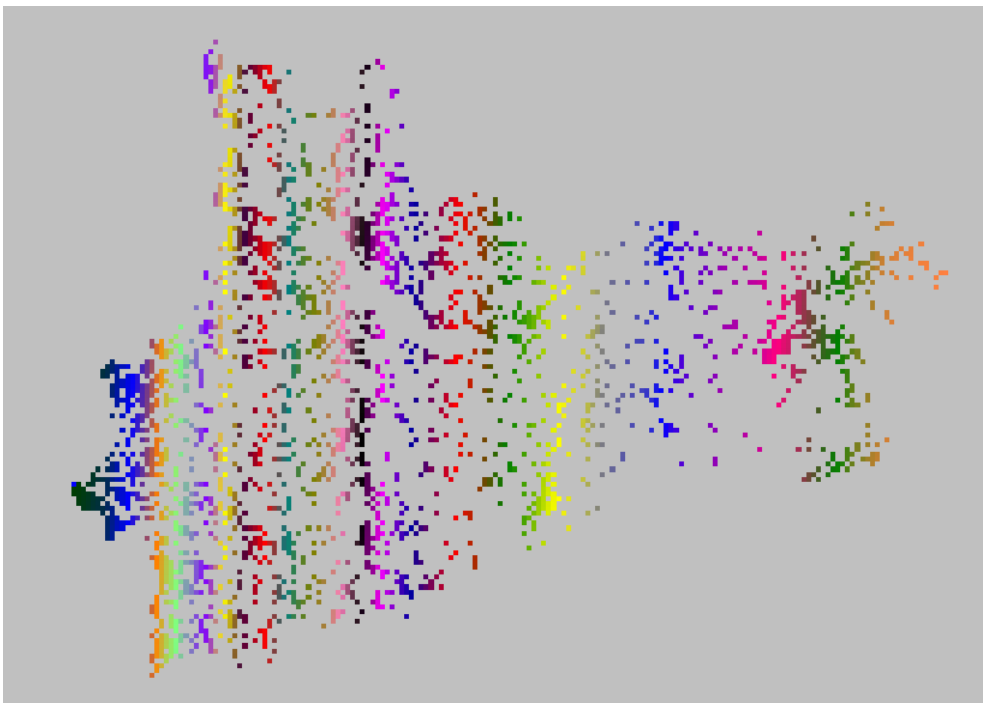


figure 2.26.3.2: HRU master cells (basin of the river Aare up Berne, Switzerland, 2426 cells

The vertical stripe-pattern is a result of the HRU creation algorithm that works column by column in order to get results that are best suited for later parallel processing with OpenMP and MPI.

Another example for HRU creation is the flat arctic tundra basin shown in figures 2.26.3.3 and 2.26.3.4 resp. Since the area is quite flat, no elevations are taken into account when creating HRUs. Only soil types and land cover are regarded. Since there are only four types for each of these

parameters (and they happen to share the same spatial distribution), the number of HRU master cells is only four. The speed-up is tremendous: pre-spin-up takes only 25 seconds for a 5 year period in hourly resolution and the spin-up itself takes about 2.5 hours for 50 years compared with several hours (about 10 or more) run time for a single year in normal model mode (speed-up factor > 200).

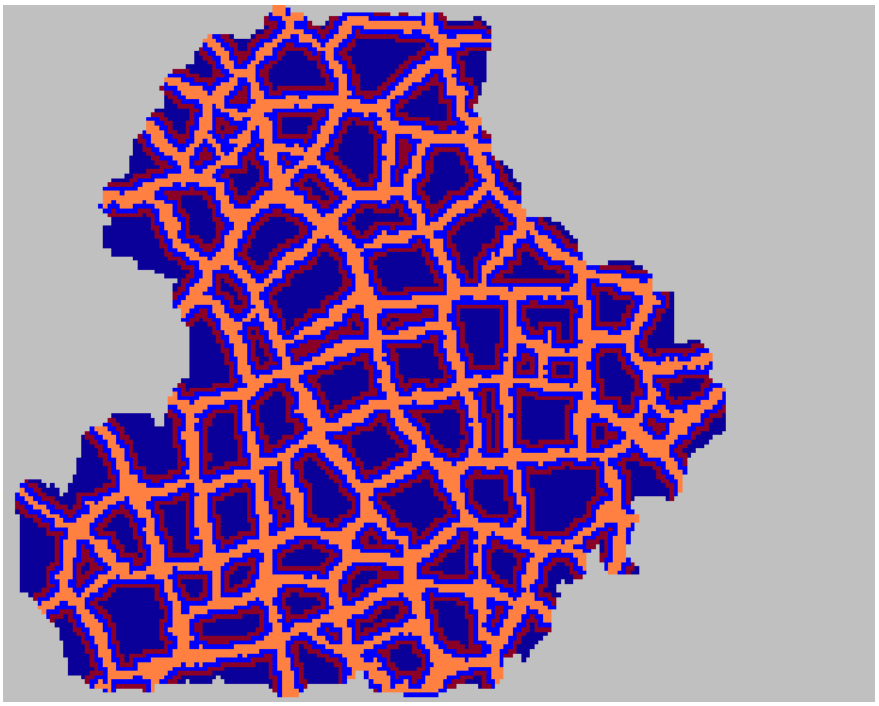


figure 2.26.3.3: All HRUs for a flat arctic tundra basin with typical permafrost structures (low centered polygons)



figure 2.26.3.4: HRU master cells for a flat arctic tundra basin

2.26.4 Examples for spin-up results including soil temperature pre-spin-up

The spin-up procedure include a procedure to construct a first guess temperature profile. This is done in the pre-spin-up run. First the mean annual air temperature is calculated by only running the air temperature interpolation for a few years (the defined model period in section [model_time]).

Following Kudryavtsev *et al.* (1974) a single equation estimates the mean permafrost table temperature from air and surface properties,

$$\overline{T}_s = \frac{K_t \cdot n_t \cdot TDD_a - K_f \cdot n_f \cdot FDD_a}{K_f \cdot p} \qquad \textbf{(2.25.1)}$$

where n_f and n_t are the freezing and thawing n-factors respectively dependent on the snow properties and vegetation,
TDD_a is the thawing degree day factor of the air (will be accumulated for positive temperatures only),
FDD_a is the freezing degree day factor of the air (will be accumulated for negative temperatures only but with a positive sign, i.e. $FDD_a > 0$),
p is the period (in days) and
K_t and K_f are the thermal conductivities of thawed and frozen ground respectively.

Organic soils will have a greater thermal offset because the water holding capacity is greater and the thermal conductivity of the organic material is low resulting in the larger difference between thawed and frozen thermal conductivities of the active layer.

The temperature at the bottom of the active layer can then be used to linearly extrapolate the temperature profile based on the specified geothermal heat flux at the bottom of the profile. Note that phase change is not important, because these are equilibrium mean annual conditions. Heat flux is only dependent on the temperature gradient and the thermal conductivity, which is depended on the frozen or thawed state.

The first guess profile is then used in the spin-up procedure where the temperature profile is modified by running the temperature model through the first few (e.g. ten) years of the data set repeatedly for about ten cycles.

An impressive example of the efficiency of the pre-spin-up for initializing soil temperatures is shown in figure 2.26.4.1. Of course, this somehow extreme example was deliberately chosen for demonstration purposes, but if the modeler does chose the wrong starting day, the same thing may happen as well during the calibration work. The straight dark blue line to the very left shows the initial soil temperature profile as created during the first model time step from the air temperature (-25°C) and the lower boundary condition (-10°C). The values in-between were linearly interpolated. After running the 5-year hourly air temperature input in normal spin-up mode for 50 years (which took about 2.5 hours on a 6-core CPU for only the HRU master cells), the soil temperature profile did finally approximately meet the temperature profile as found after the pre-spin-up (25 seconds on the same CPU) and a following normal 5-year spin-up (15 minutes).

This example is not an extreme example at all. For basins with a wide range of elevations, e.g. ranging from about sea level to several thousands meters, the temperature range is quite wide. To initialize the high mountain permafrost temperatures more realistic, this spin-up mode including a pre-spin-up for soil temperatires is very effective and highly recommended.

As can be seen in figure 2.25.8.12, the temperature profile using the pre-spin-up (dashed lines) does not use the same initial lower boundary condition (which was a constant temperature of -10°C for

the version without spin-up) but rather a geothermal heat flux (65 mW/m^2 in this example). The temperature profile is estimated by assuming saturated conditions below the permafrost table for estimating the effective thermal conductivity.

The sharp bent in the uppermost soil layer is an artifact due to the fact that the model had to run for at least one time step in normal spin-up mode in order to generate the first temperature profile. Since that first hour was quite cold (-25°C as in the initial profile without spin-up), the temperature of the uppermost layers reacted already to the input during this first spin-up time step.

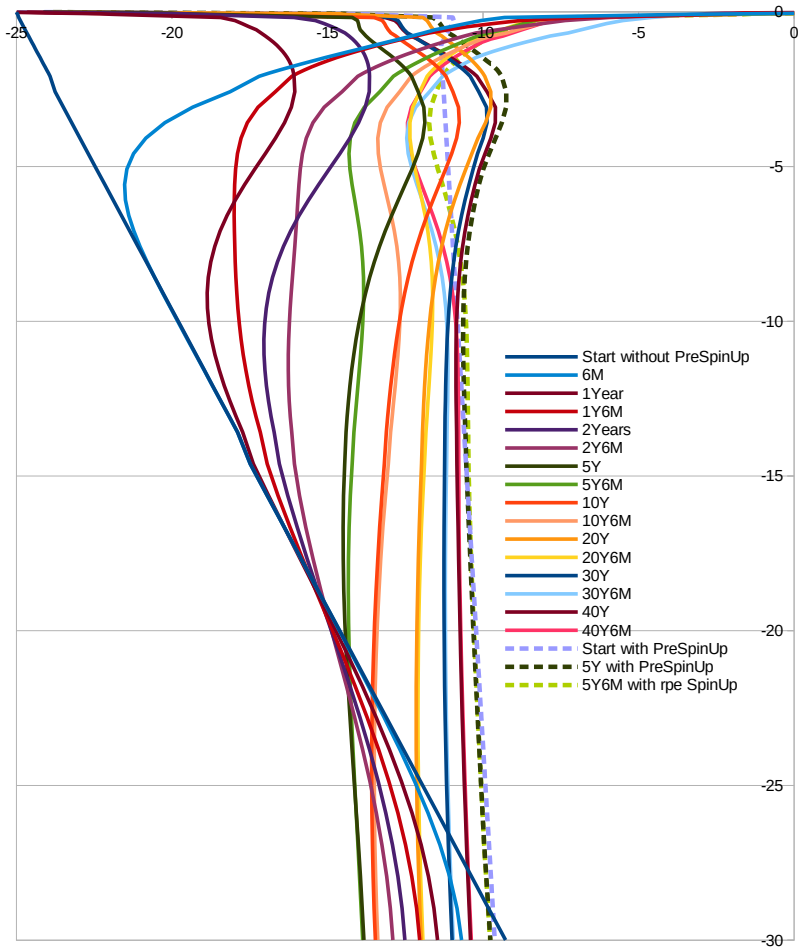


figure 2.26.4.1: soil temperature profiles without pre-spin-up (solid lines) and with pre-spin-up (dashed lines

2.27 Layered Snow model

2.27.1 General considerations

Note, that the parameters for the energy balance snow melt methods may have to be re-calibrated when using the new layered snow model, since a compact snow pack with out layered snow model reacts quite different when warming up or cooling down compared to the layered snow pack. The layered snow model is available for the Richards-version only and runs with activated heat transfer only.

The layered snow model is the main new feature in version 10.00.00 of WaSiM. Here, the snow is looked at as a stack of additional "soil" layers, i.e. the "soil" can grow in thickness during winter season by adding layers bottom up on top of the soil and shrink by removing layers top down during melt season. The main reason to add such a complex new feature was to be able to model the soil temperatures more realistic, since the actual method of using land use or temperature dependent factors (the so-called n-factors) is often not satisfactory and also not physically based at all. With a layered snow model, the heat transfer can be modeled for the snow pack, so the surface temperature for the soil temperature model is simply the value for the soil surface resulting from the continuous heat transfer through snow and soil. However, this holds for the snow covered time only, since for snow-free times the algorithm works as before: No snow – no additional "soil" layer – the surface temperature is taken again from the air temperature and an n-factor can be applied. There is actually no energy balance implemented for snow free times that would provide a surface temperature similar to the snow melt method with energy balance approaches. One main advantage of modeling snow temperatures is to get the time offset for soil warming and permafrost thawing during the melting season much better than before, when the snow cover had no impact at all on the surface temperature of the soil. That can make a big difference in the final thaw depth of permafrost soils and in the long term thermal regime, especially when the snow cover is quite thick like in higher alpine regions.

The idea behind the implementation is to dynamically add "soil" layers to the soil model that will then be modeled with the existing algorithms for water and heat transfer (so that also means that no additional equations need to be listed here because everything is already there, see chapters 2.14 for water fluxes and 2.25 on heat transfer). The snow model is part of this new layout: Accumulation of snow or sublimation of ice is always done in the snow model. Also, if a new snow layer has to be added, this is done in the snow model. When using simple melt approaches like the T-(u)-index approaches (and also simplified energy balance methods 3 and 4 after Anderson), the melt is also subtracted from the snow layers within the snow model already. For all (true) energy balance methods the melt will be computed implicitly when running the heat transfer model in the unsaturated zone sub-model (only available in Richards-version of course). The only additional input (beside solid/liquid SWE values and temperatures for all snow layers) that is needed from the snow model is the energy balance in W/m^2 which is a result of the energy balance melting approach. So instead of melting the snow directly in the snow model, the energy balance is passed to the unsaturated zone model and there a virtual surface temperature is calculated (ΔT corresponding to the heat flux that is equivalent to the passed-on energy balance) that is then used as surface temperature for the heat transfer model. The energy balance itself can not be used directly, because the heat transfer model is driven by temperatures as boundary conditions.

One intrinsic problem of this method (of using snow as additional soil layers) is that the snow is the soil matter itself and that the snow can change its aggregation state to water and vice versa depending on the layers "soil" temperature. So the snow is both, "soil" substrate and pore filling

water. That is also the main physical/technical difficulty in the actual implementation. The key connection between the solid and the liquid phase of soil water is the temperature and, connected with it, the relative saturation (that depends on temperature and van-Genuchten parameters). If the snow warms up near the melting point, which is not at 0°C since we have a negative capillary pressure that moves the melting point to negative temperatures like for real soils, the relative saturation rises and more water can flow freely and leave the layer. However, that means that also the soil substrate melts away. The actual implementation keeps the snow density constant (may change in later releases), so the thickness of the layer must be adopted to the remaining solid snow after a certain amount of snow has been molten. As a direct consequence of the fact that the relation of solid and liquid water in these snow layers (or the relative saturation SE) depends on the snow temperature, a correction for the snow temperature was implemented that is always applied after any liquid water left the snow layer in order to keep the absolute amount of liquid water in a snow layer constant. The removal of water leads to a lower relative saturation and to bring this into equilibrium again, either additional solid snow needs to change its phase to water (leading to even more runoff and thus to a positive feedback loop and more and more melting snow) or the temperature must be adopted to meet the actual relative saturation again. The latter approach is implemented in WaSiM, so during melt periods, the temperature may stay at somewhat deeper levels than expected (usually only some 1/10-th of a degree). This also depends on the chosen van-Genuchten parameters (remember: snow is a soil layer, so all parameters that describe the water and heat transfer must be defined for snow as well).

2.27.2 Technical aspects

WaSiM stores all soil properties in stacks of grids. Each numerical layer is such a grid. A soil with e.g. three horizons and several numerical layers in each horizon may have a layer count of 20 to 30 or even more numerical layers. Such a stack is created for several entities:

- hydraulic head (hhyd)
- water content (theta)
- geodetic elevation (hgeo)
- fluxes between soil layers (qu)

When using heat transfer, an additional stack is added:

- soil temperatures (tsoil)

Now, when using the layered snow model, even more additional stacks are required:

- hydraulic head (hhydsnow)
- water content (thetasnow)
- geodetic elevation (hgeosnow)
- fluxes between soil layers (qusnow)
- snow layer temperatures (tempersnow)
- snow layer densities (denssnow)
- snow layer thicknesses (thicksnow)
- snow layer water equivalent solid (swes)
- snow layer water equivalent liquid (swel)

In addition to the large number of stacks, the number of snow layers should be big enough: if there are parts of the basin that accumulate lots of snow, the number of layers must be big enough to cover the highest expected snow fall. Though the uppermost layer may contain an unlimited amount of snow (to keep the balance correctly), the heat transfer model will not work correctly with very

thick snow layers. So it is reasonable to define e.g. 30 layers of snow with a maximum thickness of 0.3 m each and a default density of 400 kg/m^3. That provides storage for 120 mm SWE in each layer, so the total snow cover may contain 30 x 120 mm = 3600 mm. The direct consequence of this large amount of layers and stacks is, that the memory requirement can be quite high.

Also, the additional stacks need more disk space. And they must provided as initial values in the StateIni-folder when running continous model periods or loading initial states from a warm up run.

Last not least, the computation time can be considerably longer than without layered snow model. But this is because of the heat transfer model that needs to be part of the layered snow model. The layered snow model is implemented in a way that configurations without layered snow model should run with the same performance as before.

2.27.3 Sample results

By adding a file with typical heat transfer model outputs to the section [special_output] it is possible to create time series of layer temperatures like the one shown here in figure 2.27.3.1. The definition of the output file may look like the following example:

```
outputfile { header = temperature_Soil;
      filename = $outpath//special_output_Temperature_Stack_Soil.//$code//$year;
      entity {
            ID     = Soil_Temperature_Stack;
            Symbol       = Tsoil;
            Xcoords      =  691750  ,691750  ,691750  ,691750  ,691750  ,691750  ,691750
,691750 ,691750 ,691750 ,691750 ,691750 ,691750 ,691750 ,691750 ,691750 ;
            Ycoords      =  164250  ,164250  ,164250  ,164250  ,164250  ,164250  ,164250
,164250 ,164250 ,164250 ,164250 ,164250 ,164250 ,164250 ,164250 ,164250 ;
            Level  = 0      ,1      ,2  ,3      ,4      ,5  ,6      ,7      ,8      ,9
,10     ,11     ,12     ,13     ,14     ,15;
      }
}
```

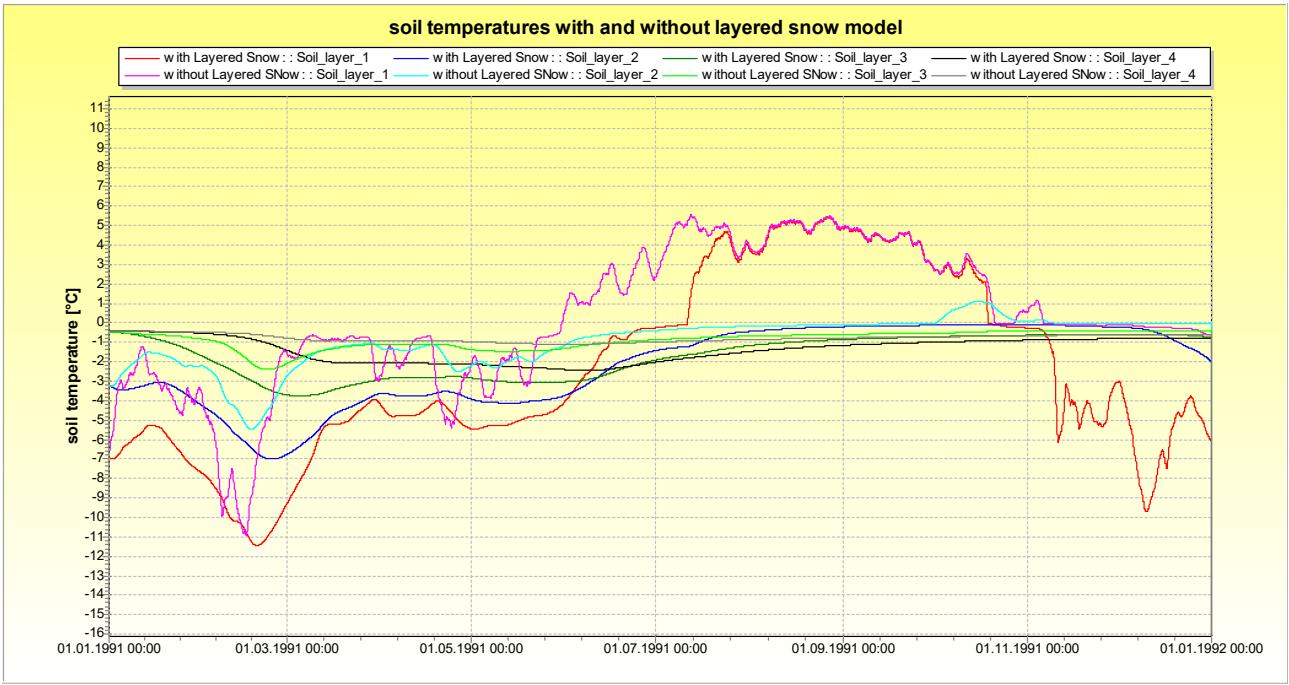


figure 2.27.3.1: soil temperatures for the first four soil layers (each 0.33m thick) with and without layered snow model

As can be seen in figure 2.27.3.1, the first soil layer starts to thaw more than a month later with

layered snow model compared to a run without layered snow model (red series for layered snow model vs. magenta for temperature without layered snow model). Also, the second layer never starts to thaw in the version with layered snow model (dark blue series) compared to the short thawing episode in the version without layered snow model (light blue series). That means, that the thaw depth is less for the version with layered snow model, since the snow cover insulates the soil from higher temperatures. Such impacts can be quite important when modeling long term permafrost simulations in high alpine or arctic regions to estimate the impact on global warming on the permafrost stability. Figure 2.27.3.2 shows the comparison between the results of a run with and a run without layered snow model. The red series is the average thaw depth when using the conventional snow model (without heat transfer model in snow), the blue series is the result when using the layered snow model. It must, however, be said, that both versions were ran with the same n-factors. By calibrating the n-factor for the run with non-layered snow model, thaw depth may be in a similar range.

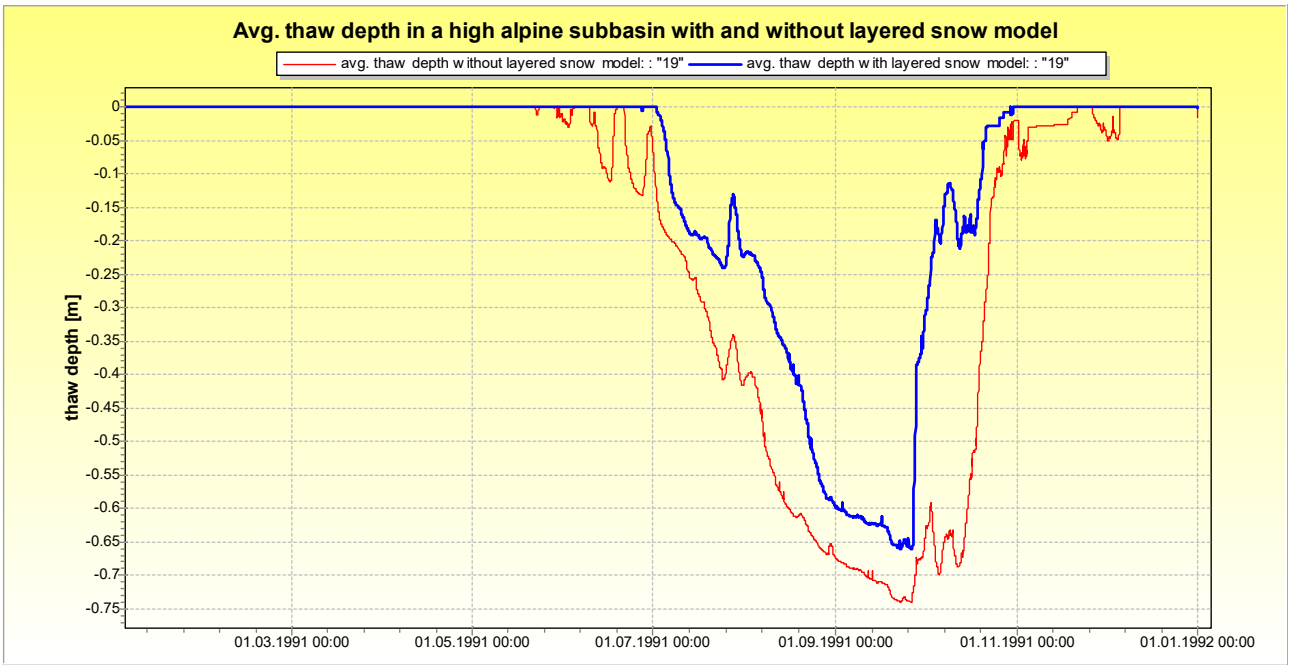


figure 2.27.3.2: average thaw depth for the same high alpine sub basin using layered snoe model (blue) and using the conventional snow model (red)

The Layered Snow model allows the output of detailed layer information on SWE via the [special_output] or via a stack. Figure 2.27.3.3 shows an example of a special output time series for SWE solid and liquid for several layers of the layered snow model. Layer 0 always contains the sum of all layers, so the fat red series is the total solid SWE, while the other series that start to grow up to around 200 mm SWE are the SWE values for single layers. Figure 2.27.3.4 shows a zoomed-in detail of the layers SWE values for the first four snow layers during the melting season. The thin lines refer to the liquid water content, the fat lines to the solid ice content (i.e. water and snow). Here, the maximum water content for a layer was defined at 50%. As soon as the water cointent reaches that limit, the exceeding liquid content is handled as direct runoff and the layers temperature is then reduced to a temperature that fits the actual SE value again (0.5 in this case). It can also be observed that the total SWE of a layer makes a positive jump when the overlaying layer melts. That is due to infiltrating melt water from that layer which is refreezing in the lower layer.

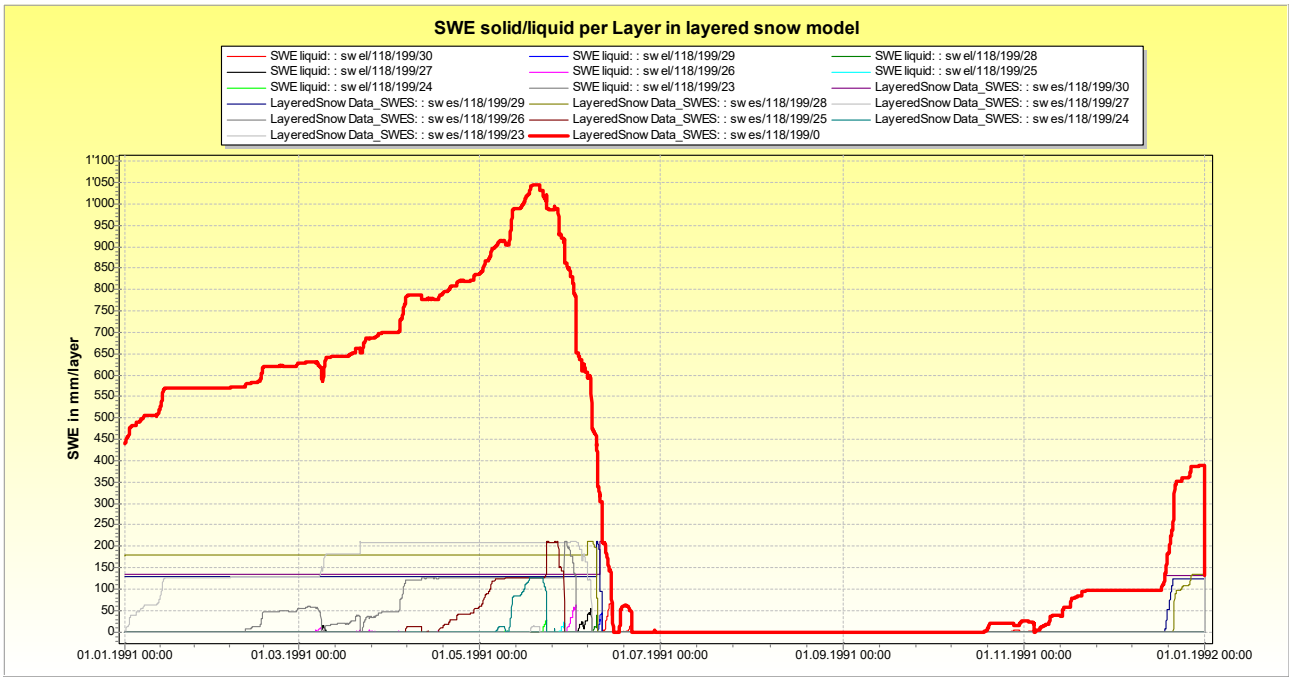


figure 2.27.3.3: SWE solid and liquid per layer as special output time series

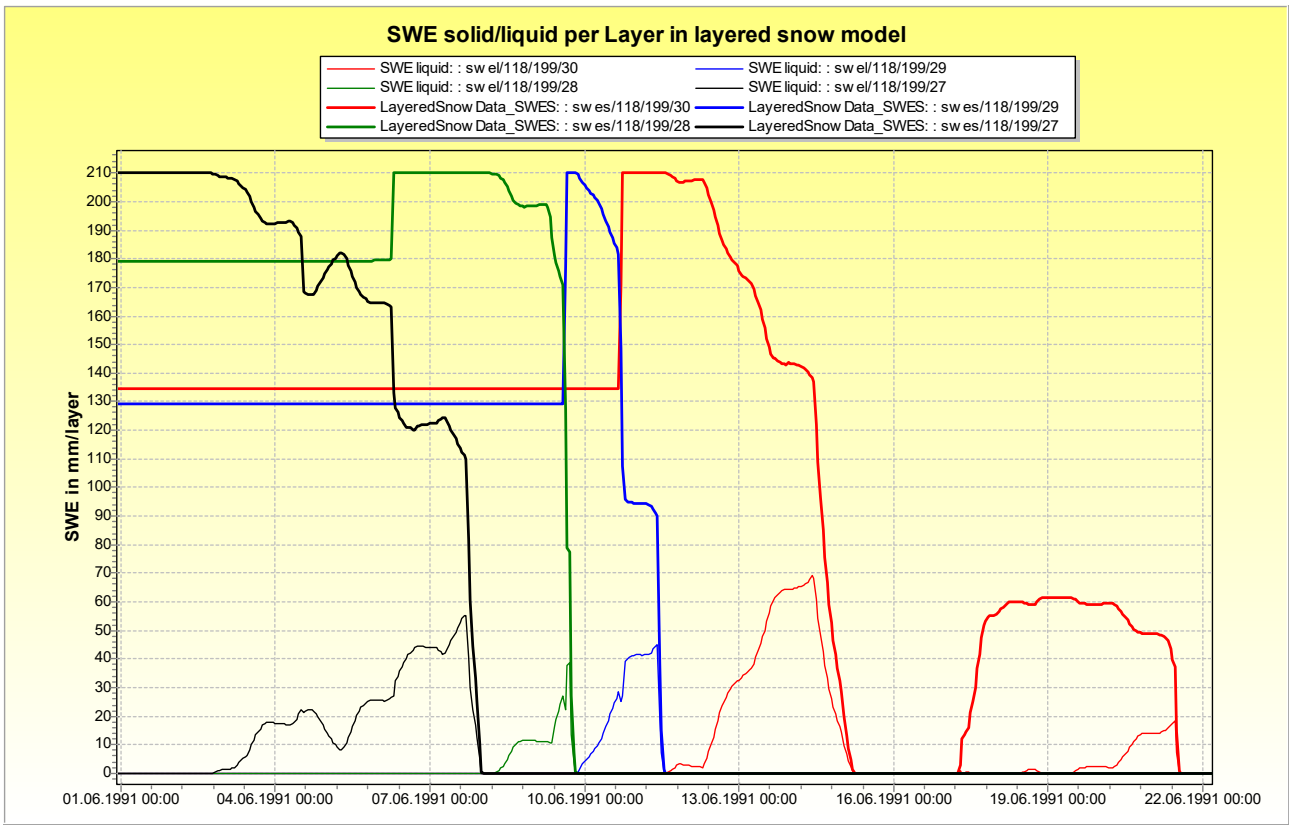


figure 2.27.3.4: Detail of SWE solid and liquid per layer. The thin graphs are the liquid water content.

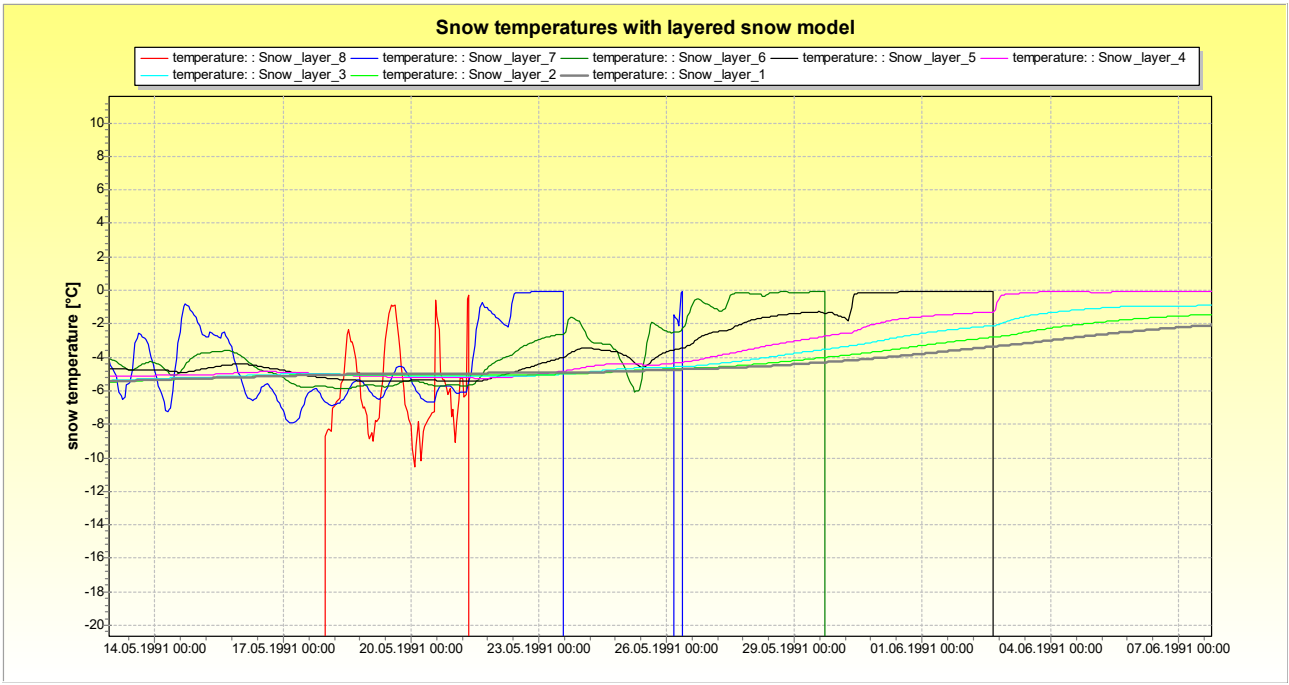


figure 2.27.3.5: snow temperatures for a melting snow pack with up to eight snow layers

Figure 2.27.3.5 shows a detail of snow layer temperatures for the melting period. The damping and time shift effect due to the heat transfer is nicely shown. (the straight lines when a layer is created or removed are atrtifacts of the viewing software). As can be seen for layer 8 (red series), a quite thin layer of fresh snow, the temperature changes during the day are quite large, while the changes for the lower next layers (blue, then dark green, then black) are much smaller and shifted in time.

2.27.4 Control file changes for layered snow model

There are two new sections required for using the layered snow model

- to define general model parameters: section [LayeredSnowModel]
- to define the properties of the snow as a soil type: section [LayeredSnowProperties]

```
# Layered Snow model is a component that can be used within unsaturated zone model. It
handles the snow like soil layers, so heat transfer and water movement will be computed
as in a soil.
# therefor it is required to have a section [LayeredSoilProperties] in the control which
defines the thermal and hydraulic prperties of snow (like a soil type)
[LayeredSnowModel]
1                 # do_LayeredSnow: 0=do not integrate snow pack into the soil model;
1=integrate snow pack into soil model, using several layers of snow (supports heat
transfer and Richards-approach of soil/snow water movement)
1             # LayeredSnowMethod: method of dealing with snow layers: 1 = no density
modelling (density is constant)
0.1           # --> not yet active: LayeredSnowMinDensity: minimum density of freshly
fallen dry snow (will later be modeled in more detail, i.e. depending on temperatur and
wind or so)
0.85          # --> not yet active: LayeredSnowMaxDensity: maximum density of old snow
(will later be modeled dynamically depending on snow age, temperature and water content
etc.)
-2.0          # recommended: -2.0 ... 0.0 temperature difference of freshly fallen snow
to Tair (because snow comes from higher atmospheric layers, the actual snow temperature
may be lower than the air temperature in 2m elevation -> otherwise snow fall at 2°C
weren't possible). This parameter leads to a cooler snow pack and less melt when the
snow pack starts to accumulate (and soils are still warm as well)
0.0           # scaling parameter for snow internal runoff (like interflow, in analogy
```

```
to drainage density in interflow module) --> use with care or let it be at 0, only
effective if k_recession < 1
0.5           # maximum water storage capacity of snow (this parameter exists because
snow is looked at as a soil type, so the soil matrix may change into water due to
warming up and become "soil moisture" itself). To make the numerical solution more
stable, this parameter should be used (0.1...0.2)
1.0           # --> set always to 1.0, otherwise unpredictable behavior! SE
effectiveness for snow: effectiveness of SE on reduction of hydraulic conductivity and
suction for snow. When set to 0, hydraulic properties of SNOW only are like without snow
model (only heat transfer uses SE then). When set to 1, the effectiveness is 100% like
in older WaSiM versions. Range 0...1. Recommended: 1
0.0           # SE effectiveness for soil: effectiveness of SE on reduction of hydraulic
conductivity and suction for soil. When set to 0, hydraulic properties of SOIL only are
like without snow model (only heat transfer uses SE then). When set to 1, the
effectiveness is 100% like in older WaSiM versions. Range 0...1. Recommended for first
using layered snow model when the calibrated model didn't use heat transfer yet: 0, else
1
# some stacks grids that may be of interest
$outpath//$SWESstack # stack with SWE solid values per snow layer
1//$Writegrid      # 0, if no grid-output is needed, else one of the codes described
above
$outpath//$SWELStack # stack with SWE liquid values per snow layer
1//$Writegrid      # 0, if no grid-output is needed, else one of the codes described
above
$outpath//$temperSnowStack  # stack with snow temperatures per snow layer
1//$Writegrid      # 0, if no grid-output is needed, else one of the codes described
above
$outpath//$thetaSnowStack   # stack with theta values per snow layer
1//$Writegrid      # 0, if no grid-output is needed, else one of the codes described
above
$outpath//$hhSnowStack      # stack with hydraulic heads per snow layer
1//$Writegrid      # 0, if no grid-output is needed, else one of the codes described
above
$outpath//$hgeoSnowStack    # stack with geodetic elevation per snow layer (bottom of
layer)
1//$Writegrid     # 0, if no grid-output is needed, else one of the codes described
above
$outpath//$quSnowStack      # stack with (bottom) water fluxes per snow layer
1//$Writegrid     # 0, if no grid-output is needed, else one of the codes described
above
$outpath//$thickSnowStack   # stack with thickness of each snow layer
1//$Writegrid     # 0, if no grid-output is needed, else one of the codes described
above
$outpath//$densSnowStack    # stack with density of each snow layer
1//$Writegrid     # 0, if no grid-output is needed, else one of the codes described
above
$outpath//$upperIndexSnow   # grid that holds the index of the first (uppermost) layer
that has a positive SWES value, i.e. the first layer with snow in it
1//$Writegrid     # 0, if no grid-output is needed, else one of the codes described
above

# snow layer properties: this is a short soil table, since the snow layers are handled
like a soil in the model
# all entries have the same syntax as for the soil table, However, some values may not
be used internally or even changed dynamically, see comments
# actually, ONLY ONE layer is really used, though.
[LayeredSnowProperties]
1 # number of entries --> ATTENTION: do not change until the model can use multiple snow
types
1 Snow { method = MultipleHorizons;
 ThicknessScaling   = 1.0 ; # use with care - effects on SWE may be unpredictable when
changing this parameter value
 PMacroThresh= 0 ; # better to use no macropores for snow (not implemented anyway)
 MacroCapacity = 0 ; # better to use no macropores for snow (not implemented anyway)
 CapacityRedu= 0 ; # better to use no macropores for snow (not implemented anyway)
 MacroDepth  = 0 ; # better to use no macropores for snow (not implemented anyway)
 EvapMaxDepth= 0 ; # there is no evaporation from snow that counts as soil evaporation.
 horizon     = 1 ; # ATTENTION: do not use more than one horizon for snow! The layers
will be filled up dynamically from bottom to top and also some of the properties will be
changed by the model itself. So one horizon is sufficient for now
 Name        = Snow ;
```

```
 DryHeatCapacity    = 2000 ; # dry heat capacity in J/(Kg*K) (This needs to be checked,
I used Cvol/density)
 DryDensity  = 400  ; # dry density in kg/m^3
 DryThermalConduct  = 2.33 ; # dry thermal conductivity in W/(m*K) (or J/(m*s*K) --> 1J
= 1Ws)
  KMinFrozenSoil    = 1e-6;    # minimum hydraulic conductivity in m/s when the soil is
coimpletely frozen -> should be quite high for snow
 ksat        = 1e-4; # like a highly porous sand or small gravel
 k_recession = 1.0  ; # although the saturated hydraulic conductivity decreases, there
will be no interflow, only increaed storage in the lower layers
 theta_sat   = 0.7  ; # (should be 1.0) this will be dynamically adjusted (at least
later, wehn the dynamic snow layers are implemented completely)
 theta_res   = 0.1  ; # (should be 0.0)
 alpha       = 18   ; # van Genuchten alpha like a highly porous sand or small gravel
 Par_n       = 4    ; # like a highly porous sand or small gravel
 Par_tau     = 0.5  ; # like a highly porous sand or small gravel
 thickness   = 0.3  ; # this can be changed if required (measure in meters). Do not use
too thin layers, since the performance will go down dramatically then (recommended
minimum 0.1m, better use 0.2 m)
 layers      = 30   ; # this can be changed if required (1...inf)
}
```

2.28 Data Assimilation

2.28.1 Problem description

Data assimilation means the incorporation of external, observed data into the running model, that can be used to adjust internal model states. Driving data like precipitation etc. are not covered by this approach. These external data are state data that usually describe storage contents of various types like reservoir contents or snow water equivalents (WaSiM uses some more storage's like groundwater tables, interception storage, soil water storage, glaciers components, water flowing in river channels or staying on the surface in ponds etc., but reservoir content and SWE are the most important ones that can be read in from external sources). The modeled states will usually differ slightly from the real states, but most storage's are kept between well known lower and upper limits, thus often providing a kind of auto-reset for the bias between model and reality, e.g. if a lake or reservoir fills up completely this state is met eventually by the observed values, but usually not exactly at the correct (model) time but some time before or after the model computed that event. So the model is synchronized automatically with the observations once the modeled state of the full reservoir equals the observed state. However, this works well for certain entities and time horizons, but not for all. For instance, the calculated snow melt from accumulated snow may differ substantially from the observed snow melt if the modeled accumulated snow water equivalent (SWE) differs substantially from the real value. This can happen due to small errors in precipitation and/or temperature interpolation as well as by snow melt routines that do not absolutely correct model the snow melt (because e.g. the radiation balance can not be estimated correctly because of missing cloudiness data etc.). Also, SWE does not have an upper limit, so there is only an automatic synchronization when all snow is molten. But the important model output of the snow melt rate is then still biased. Also, big reservoirs or lakes are often not filling up during a single year, some large lakes, especially in arid climate zones, do even not fill up in decades, so the worst case is, that there is no automatic synchronization at all if these lakes do not even have any outflow. One of the most popular examples may be Lake Tchad in the Sahel zone in Africa, which grows and shrinks considerably every year without ever falling completely dry and without ever filling completely up (a calibration for this lakes basin was made by the author, using quite sparse precipitation input data for 16 years but by using observed water tables as well – still without direct data assimilation). Also, snow accumulation areas of glaciers in high elevations that do not melt completely are subject to accumulating small differences in modeled to real SWE. A special example are modeled SWE values in high alpine regions, where modeled snow fall is often too high and modeled snow melt is

much too small, so the snow accumulates over the years to tremendous amounts. One approach to limit these snow packs is to use a snow slide approach – which is provided by WaSiM with the extended snow melt approaches. Another approach, however, could be the replacement of the modeled SWE values by observed values by using the new data assimilation techniques.

It the model should still provide realistic outputs from such biased SWE accumulations, and a better calibration of modeled SWE is not possible (because of lacks in comprehensive input data or even in the physical description of the process), an alternative method is to adjust the internal states from time to time with observed ones, so that the accumulated errors are reset. This is called data assimilation.

Starting with version 10.04.04 WaSiM provides two possibilities to use external data for data assimilation purposes:

- Grid-type data assimilation, the external data is provided for the entire model domain as gridded data. The implementation allows a simple grid type data import or an import with special snow water equivalent options (see below)
- Reservoir-type data assimilation: for lakes or reservoirs, the observed water table can be used to adjust the modeled water content of the reservoir.

2.28.2 Grid-type data assimilation

Each known grid can potentially be read in by using this method. Only a grid identifier and a list file containing date-time values and grid names are required (see control file specifications below). However, if there are dependent entities, the method may fail if only the base grid is read and the dependencies are not actualized. Actually, some special options are implemented for snow water equivalent (SWE) only. Since the snow model knows not only the total SWE but also liquid and solid fractions, these entities need to be set after reading a new SWE grid. Also, if the glacier model is switched on, all three grids (SWE-total, SWE-liquid and SWE-solid) must be set for the glacierized fractions of glacier cells as well.

If the cell contains already a snow pack, the solid and liquid fractions are set based on the actual fractions. However, if the last grid value was zero, the fractions of liquid and solid SWE fractions is set by using a default SWE-liquid fraction.

One important extension in the actual version is that there is no need for any input grid to match the sub-catchments grid dimensions. All grids will automatically be interpolated to the master grids dimensions and cell size. However, the method for interpolation is always bi-linear interpolation, as is the case for imported GIF and netCDF grids (the method called for interpolation is the same as for netCDF and GIF files). So this method should not be used for categorical data like land use, if the read in grids have other dimensions or cell sizes or coordinates than the internal model grids, otherwise the interpolated cells will contain invalid values. All continuous data, however, can be interpolated without restrictions (may be the only restriction is that all model cells should be enclosed by the read in grid, because the bi-linear interpolation delivers feasible results only for locations that are between at least four valid input locations.

2.28.3 Reservoir-type data assimilation

A reservoir-type data assimilation uses a single water level as input value and updates that reservoirs content according to the waterlevel-volume-function. In order to be able to account for systematic offsets in the modeled and the observed water tables, the observed average water table must be provided as a parameter. The observed values should be provided in a WaSiM-table format file. Each time, a new observed value is provided, the lake model will use the read in value and

compute the actual content of the lake. For all cells that belong to the lake, the pond values will be re-computed and the reservoir statistics (i.e. water table, content, area and all tracer concentrations) are updated.

A reservoir type data assimilation can be set for a single reservoir only. However, it is possible to define reservoir-type data assimilation rules for as many reservoirs as required.

2.28.4 Control file extensions for data assimilation

The following section shows an example how the control file should be extended to use data assimilation.

```
[data_assimilation]
1 # 0 for no data assimilation, 1 for running data assimilation
$time # the general model time step
2 # rule count
grid { grid-ID = snow_storage_total;
       listfilename = d:\Data\SWE_observed\observed_SWE.list;
       options # here some special options for SWE grids
       {
            split-into-liquid-and-solid         = true;
            weighing-by-glacierized-fraction    = true;
            default-liquid-percentage           = 5.0;
       }

}
reservoir { lake-ID = 43;
      AverageWaterTable = 480.2;
      listfilename = d:\Data\lake-levels\lake43.asc;
      options
      {
            format      = wasim; # actually the only officially supported format
            skiprows    = 23; # how many rows should be skipped if not for every
# time step the data assimilation should be done, e.g. 23 if time step of the
# model as well as of the input data is one hour and we need a synchronization
# once per day only
      }
}
```

The key for the new (optional) control file section is [data_assimilation], the first two lines are the common parameters as used for all sub models: the general do-switch (0=no, 1=yes) and the common model time step.

The next line contains the number of following rules, here two. For each rule, a section of the syntax

<type> {<parameters> [options {<options>}] }

must be defined. <type> can be either **grid** or **reservoir**.

For **grid-type** data-assimilation, the general parameters are the internal grid-ID (i.e. the name under which the grid can be found, see the list at the and of this manual) and the file name of the file that contains the names of the files to load together with their respective time in WaSiM table format (see interpolation method 9 for externally interpolated grids for a specific description). The difference to the usual WaSiM input file is, that the date and time for each entry is arbitrary, i.e. it is not required to have an entry for each model time step. The only requirement is, that the date-time-entries are sorted correctly.

There are several options that were implemented for snow water equivalent data assimilation:

- split-into-liquid-and-solid: when SWE grids are imported, the internal WaSiM grids for liquid and solid states need to be updated. This switch allows the feature to be switched on or off
- weighing-by-glacierized-fraction: should be set to true if the glacier model is used
- default-liquid-percentage: the default value for liquid and solid SWE fractions when there is actually no snow on the cell

For **reservoir-type** data assimilation, the lake-ID and the AverageWaterTable are the most important parameters beside the list file that contains the observed water tables. This file is a file in the well known WaSiM table format. As for grid-type data assimilation, the dates are arbitrary, so only when a new observation value is present, the data assimilation is done. There are two possible options: the format of the input file (actually only „ wasim‟ is supported) and skiprows, a parameter that defines how many rows in the input list file should be skipped after each assimilation step.

2.29 Modeling Ice Wedge degradation and Changes in the arctic Vegetation with ArcVeg

2.29.1 Basic concepts of ArcVeg

Ice wedges in arctic tundra regions and their degradation can have a major impact on the local hydrology. The ice wedge itself blocks any lateral flows whereas the degradation leads eventually to the creation of an effective drainage system that can dramatically change the water budget of a landscape. WaSiM implements an ice wedge degradation model that models the creation of those channels by decreasing the soil thickness of all soil layers that are part of an ice wedge.

Concept:

- The ice wedge degradation is part of the ArcVeg implementation and can't be used without ArcVeg (however, it is possible to set the arcveg parameters to such values that nothing will happen except ice wedge degradation).
- The original Ice Wedge location and depth (upper and lower limits) are defined in two new parameter grids (ice-wedge-upper-limit and ice-wedge-lower-limit).
- A new Excess ice stack stores the amount of ice (in mm WE) for each soil layer.
- The ice-wedge-upper-limit grid defines for each cell if it is on an ice wedge (otherwise: nodata) and what is the upper limit (in m below ground surface)
- The ice-wedge-lower-limit grid defines the lower limit of the ice wedge (in m below ground surface)
- Initialisation mode: If a cell lies on a valid cell in the ice-wedge-upper-limit and ice-wedge-lower-limit grids (i.e. it's not a nodata-cell in these grids), then all excess-ice-layers (correspond to soil layers) that are between the upper and lower limit of the ice wedge will be assigned an amount of ice that equals the layers thickness. Since the excess-ice-stack has a measure of mm WE (water equivalent), the thickness must be linked to WE by the density of ice, which is assumed to be constant at 918 kg/m^3, i.e. a factor of 0.918 is applied to the soil layer thickness to get the excess ice. That implicitely means, that the ice wedge is assumed to consist of 50% soil and 50% excess ice.
- Warm-start: the existing excess-ice-stack is loaded (and The ice-wedge-upper-limit and The ice-wedge-lower-limit grid are igored), so the degradation model can always be interrupted and reloaded like all the other storage grids in WaSiM.
- If the heat transfer model indicates any water outflow (i.e. melt) from a layer that is within an ice wedge, that mass is taken from the excess ice stack until the value is zero.
- The thickness change is applied to all layers that are part of an ice wedge and that had any runoff modeled in the actual time step (thickness change is done for each time step).
- The resulting stack of the geodetic heights (hgeo) and the DEM itself will be stored after the model run. It is important to not only read in the hgeo stack for the next model run but also the changed DEM (which is stored under the same name but with an extended extension of "_out"). That grid should be renamed to the input name of the DEM grid for any subsequent runs. The hgeo stack is NOT initialized for every model run anymore. It's up to the modeler to keep the DEM grid and the hgeo stack in sync.
- Control file changes will be described together with the ArcVeg changes below.

The adopted soil thicknesses and the changed DEM will be used as input to ArcVeg.

ArcVeg is an Arctic tundra vegetation dynamics model, originally developed (beginning in 1996) by Howard Epstein, Anthony Starfield, Marilyn Walker, and Terry Chapin to interact with a caribou energetics model as part of an NSF-funded human dimensions project on the "Sustainability of

Arctic Communities" (Epstein et al. 2000, Kruse et al. 2004). The model has gone through numerous iterations and applications since its origin; the version of ArcVeg that was used to link with the WaSim model was coded in Matlab and re-coded to integrate with WaSim by Jörg Schulla (completed in 2025).

The basic premise of the model is that it runs on nitrogen mass balance (nitrogen is a limiting plant nutrient in Arctic tundra) with difference equations moving nitrogen among three main pools: soil organic nitrogen, plant-available nitrogen, and the nitrogen in plants. The unique feature of the model is that multiple plant types are parameterized for their growth and survival traits, and these different plant types compete for available nitrogen, producing the dynamics in the vegetation. Five plant types are parameterized for integration with WaSim: deciduous shrubs, evergreen shrubs, terrestrial graminoids, aquatic graminoids, and mosses. Grid cells in ArcVeg are 1x1m, and all variables are expressed on a m^{-2} basis; grid cells are essentially independent in ArcVeg (although not in WaSim), as there are no cell-cell spatial interactions.

ArcVeg and WaSim essentially run in serial on an annual basis. WaSim runs first, and at the end of the vegetation growing season / water year, WaSim passes two variables to the ArcVeg code for each grid cell: growing degree days and maximum depth of surface water inundation. Growing degree days (GDDs) determine the length of the growing season in ArcVeg and the quantity of nitrogen converted from soil organic nitrogen to plant available nitrogen. Depth of surface water inundation determines the ability for each plant type to acquire nitrogen and grow under submerged or partially submerged conditions. *See below for more details.*

Following, we describe each of the ArcVeg submodules, and their integration with WaSim.

ArcVeg Driver
In the ArcVeg Drive module, the following variables are initialized:
(clim) – The climate zone for which the model is running, ranging from 1 (Subzone A [highest Arctic tundra]) to 7 (boreal forest)
(planttypes) – the number of plant types in the simulation (5 for WaSim)
(son_init) – the initial quantity of soil organic nitrogen for all grid cells
(biom) – the initial biomass of each of the plant types can be set to zero, or they can be initialized with other starting values
(seedmass) – a small amount of biomass for a new seedling (set to 0.1 g m^{-2})
(GridSize) and ***(wedgePresent)*** – the initial size of the grid and the presence or absence of ice wedges will be set in WaSim and used in the ArcVeg code

ArcVeg (main) – the main ArcVeg module reads in the ***nitrogen (N) mineralization (conversion from organic N to plant-available N) parameters*** and the ***plant type parameters***, runs the ArcVeg Driver module, and calls the following submodules on an annual basis: Climate; Inundation; Mineralize; Seeds; Growth; Senesce

Climate – the Climate module is essentially a "look-up" table, with some random probabilistic elements that converts the growing degree days (GDDs) for that year (from WaSim) to the length of the growing season (for each grid cell). The growing season in ArcVeg is split into plant growth periods (PGPs). Growing season length can range from two PGPs (shortest, coldest) to five PGPs (longest, warmest) determined by the GDDs.

Inundation – the Inundation module selects / constructs a growth scalar for each plant type in each grid cell (based on ice wedge presence and surface water conditions). These scalars are ***plant type input parameters,*** which express the individual plant type responses to growing on an ice wedge and their capacities for growing either partially or fully submerged. The scalars operate on the ***nitrogen uptake efficiency (plant type) parameter,*** which ultimately affects their abilities to grow.

Presently, >30 cm of surface water is considered to be full inundation of vegetation (this value of 30 cm is currently hard-coded but could be made into a user-defined parameter). For values of surface water inundation between 0 and 30 cm, the inundations scalars are actually linearly scaled themselves based on the actual inundation depth.

Mineralize – the Mineralize module determines the amount of soil organic nitrogen that gets converted to plant-available nitrogen in each plant growth period (PGP) by grid cell; these proportional values are in the ***nitrogen mineralization (NMin) input parameters***, and vary by climate zone **(clim)** and PGP.
For each grid cell and PGP,
Nitrogen mineralized (**nmin)** = soil organic nitrogen **(son) * NMin (clim, PGP)**

For each grid cell,
Total nitrogen mineralized by grid cell **(nmin_tot)** = sum **(nmin)** across PGPs
New soil organic nitrogen **(son)** = **(son)** – (**nmin_tot)**

Seeds – For each grid cell, if a plant type is capable of growing in the given climate zone (***plant type parameters***), new biomass (0.1 g m^{-2}) will be randomly added.

Growth – the growth module is where most of the action occurs. For each grid cell, plant type nitrogen partitioning and uptake (growth) occurs in each plant growth period (PGP).
For each grid cell and PGP,
First, the nitrogen "demand" **(need)** for each plant type is calculated as their current biomass **(biom)** * their **Nitrogen Uptake Efficiency (*plant type parameter*) * their Inundation Scalar** (from Inundation module). The total demand for all plant types **(needsum)** is calculated as **sum(need)** across all plant types.
If total demand **(needsum)** <= **(nmin)** (from Mineralization module), each plant type gets their demand **(need).**
If total demand **(needsum)** > **(nmin)**, each plant type gets an amount of the available nitrogen **(nmin)** in proportion to their demand **(need)** relative to the total demand **(needsum)**.
If there is "unused" plant-available nitrogen, e.g. **(needsum)** < **(nmin)** in a PGP, that nitrogen gets added to **(nmin)** for the next PGP; any available nitrogen left over after the final PGP is "lost" from the system.
Plant types can also "retranslocate" nitrogen (***plant type parameter***), i.e. move nitrogen from their leaves into "storage" before dropping their leaves (see Senesce module below). The amount of nitrogen retranslocated **(retrans)** for each plant is split equally across the PGPs.
Nitrogen taken up by each plant type **(uptake)** and retranslocated **(retrans)** is converted to new biomass according to each plant type biomass:N ratio (***plant type parameter***), and the new biomass is added to previous biomass: **(biom)** = **(biom)** + **Plants(biomass:N) * (uptake + retrans).**

Senesce – in the Senesce module, plants "retranslocate" nitrogen from their foliage to storage, and then drop their leaves at the end of the growing season.
For each grid cell and plant type,
Nitrogen retranslocation **(retrans)** is calculated as **(biom) * foliar proportion of biomass (*plant parameter*) * N proportion in foliage** (***plant parameter***) *** retranslocation proportion** (***plant parameter***).
(retrans) is then scaled relative to the maximum biomass for that plant type (***plant parameter***), i.e. as biomass approaches the maximum biomass for that plant type, retranslocation **(retrans)** is reduced proportionally. This is effectively to prevent "runaway" biomass.
A proportion of biomass for each plant type then senesces (i.e. dies) (***plant parameter***). **(biom)** = **(biom)** – **((biom) * proportion senescing)**
For each grid cell,

Nitrogen in the senescing biomass is transferred from the plant to the soil organic nitrogen pool **(son)**.
(sensum) = sum(nitrogen in senesced biomass) across all plant types
(son) = **(son)** + **(sensum)** – sum**(retrans** across all plant types**)**; the soil organic nitrogen **(son)** is increased by the nitrogen in the senesced biomass **(sensum)** minus the nitrogen that was retranslocated sum**(retrans)**.

Information passed to WaSim
Following the ArcVeg run for the year (growing season), ArcVeg passes the biomass of each of the plant types by grid cell back to WaSim. WaSim uses the biomass values for shrubs and graminoids in equations for land-atmosphere exchange of water (i.e. evaporation and transpiration). Moss biomass is converted to depth based on the biomass and moss density; and moss is incorporated into WaSim as an organic soil layer.

LITERATURE
Epstein, H.E., M.D. Walker, F.S. Chapin III and A.M. Starfield. 2000. A transient, nutrient-based model of arctic plant community response to climatic warming. **Ecological Applications** 10:824-841.

Kruse, J., B. White, H. Epstein, B. Archie, M. Berman, S. Braund, F. Chapin III, J. Charlie Sr., J. Eamer, N. Flanders, B. Griffith, S. Haley, L. Huskey, D. Klein, G. Kofinas, S. Martin, S. Murphy, W. Nebesky, C. Nicolson, K. Peter, D. Russell, A. Starfield, J. Tetlichi, A. Tussing, M. Walker, O. Young. 2004. Sustainability of arctic communities: an interdisciplinary collaboration of researchers and local knowledge holders. **Ecosystems** 8:815-828.

2.29.2 Coupling ArcVeg with WaSiM – practical Guide

The following section describes, how to configure WaSiM to use the combined Ice Wedge degradation and ArcVeg modelling. There are some changes to be made to the control file, some grids that must be created and some procedures to be followed in order to get proper results. This is a kind of step-by-step manual that should enable the user to use that component successfully.

Step 1: Adding a new section [ArcVegIntegration] to the control file

The following section shows the additional section that must be added to the control file.
Start Julien Day and End Julian Day are of course choosable as well as the waiting period. The number of climate zones and PGPs per climate zone as well as the mineralization is taken from ArcVegs original parameterization and should not be changed without knowing what one is doing.

AVLUSEType marks the land use type, that will be used for a soil thickness change. Here, we use moss (which is land use type 1 in the land use table).

AVLayerCnt defines, how many soil layers are assumed to be that land use type (here: the uppermost three layers are moss). These three layers will be changed in thickness when changing the moss biomass by ArcVeg. This change will then also be applied to the DEM and the stack with the elevations of all soil layers and snow layers (when using the layered snow model).

AVSoilDryDensity is the density of dry soil for the AVLUSEType. Here, we use 10 Kg/m^3, meaning, that an additional biomass production of 100 g will add 1 cm to the soils thickness, so the layers 1 to 3 will be increased in their thickness by 1/3 cm each.

All files that are referenced by variable names are, as always, declared before the section itself. The

prefix AV only refers to ArcVeg, “in” is inundation, “n” is nitrogen etc. The variables are more or less self-explaining.

```
$set $ExcessIceStack                = exice//$stack//.//$suffix
$set $ArcVegGrowingDegreeDays       = avgdd//$grid//.//$suffix
$set $ArcVegInundationAvg7DayMax    = avin7dmax//$grid//.//$suffix
$set $ArcVegInundationHelperStack   = avinhlp//$stack//.//$suffix
$set $ArcVegTemperatureDailyAvg     = avtdavg//$grid//.//$suffix
$set $ArcVegInundationDailyAvg      = avindavg//$grid//.//$suffix
$set $ArcVegSnowFreeDays            = avsnowfree//$grid//.//$suffix
$set $ArcVegPlantGrowPeriods        = avpgpmax//$grid//.//$suffix
$set $ArcVegInundationScaling       = avinscaling//$grid//.//$suffix
$set $ArcVegNmineralizationStack    = avnmin//$stack//.//$suffix
$set $ArcVegNmineralizationTotal    = avnmintot//$grid//.//$suffix
$set $ArcVegSoilOrganicNitrogen     = avson//$grid//.//$suffix

[ArcVegIntegration]
1 # do_ArcVeg; 0=no, 1=yes
# parameters
153 # start Julian Day for ArcVeg. Meanings: start day activates ArcVeg, all
grids are set to 0.
288 # end Julian Day for ArcVeg. Growing Degree Days are calculated immediately
after snow melt until end Julian Day. Inundation 7-day-average is calculated
after start day and wating period until end day
8 # waiting period in days for ArcVeg. After start day, ArcVeg waits for
complete snow melt and waits then the waiting period until Inundation is
analyzed. Growing Degree Days are calculated immediately after snow melt until
end Julian Day.
# nitrogen mineralization rates per year for all 7 climate zones for 5 plant
growing periods each
7 5 # count of climate zones, count of PGPs per climate zone
# PGP1            PGP2        PGP3        PGP4        PGP5
0.000188    0.000068    0.000008    0.000000    0.000000 # climate zone 1
0.000406    0.000146    0.000016    0.000000    0.000000 # climate zone 2
0.000120    0.000550    0.000160    0.000040    0.000000 # climate zone 3
0.000120    0.000825    0.000240    0.000060    0.000000 # climate zone 4
0.000130    0.000400    0.000820    0.000220    0.000300 # climate zone 5
0.000210    0.000625    0.001300    0.001500    0.000500 # climate zone 6
0.000210    0.000625    0.001300    0.001500    0.004000 # climate zone 7
5 # actual climate zone for this setup
# ID of the land use that is part of the soil - i.e. the MOSS layer.
# So, if MOSS has land use code 1, put a 1 in here. When ArcVeg is running,
# this landuse fractions mass change will be used for changing the thickness of
the
# uppermost n layers with n being the next parameter
1 # AVLUSEType
# how many soil layers should be regarded for thickness change. This should
usually be the first horizon using 2 or 3 layers of 10 to 30 cm
3 # AVLayerCnt layers to be changed (so that not only one layer gets thicker or
thinner
# Density of the soil to be changed in Thickness, i.e. of MOSS - this is
approx. 100g per cm thickness, so 1 m^3 of that soil would weigh 100 x 0.1 kg =
10 Kg (dry, of course)
10.0 # AVSoilDryDensity Kg/m^3 dry density of the soil type whose thickness
will changed by ArcVeg
$outpath//$ExcessIceStack    # (*) stack, factor that expresses the excess ice
in that soil layer
$Writestack                  # Writecode for this stack
$outpath//$ArcVegGrowingDegreeDays # (*) GDD for ArcVeg: is used when
calculating the vegetation fractions at the end of the growing season. Depends
on Temperatur (i.e. GDD) and avg. 7 Day max inundation
$Writegrid                  # Writecode for this grid
$outpath//$ArcVegInundationAvg7DayMax # (*) maximum value of the 7-day moving
average of the inundation, counted after waiting period after arcVeg start JD
```

```
$Writegrid                    # Writecode for this grid
$outpath//$ArcVegInundationHelperStack # (*) used to compute the moving average
of the inundation depth. Writegrid must be set to at least 3
$Writegrid                    # Writecode for this stack
$outpath//$ArcVegTemperatureDailyAvg # (*) daily average air temperature for
ArcVeg: is used for calculating the growing degree days later, only in use
during start to end period of arcveg
$Writegrid                    # Writecode for this grid
$outpath//$ArcVegInundationDailyAvg # (*) daily average inundation depth for
ArcVeg: is used for calculating the actual and the maximum 7-days moving
average of inundation later, only in use during start to end period of arcveg
$Writegrid                    # Writecode for this grid
$outpath//$ArcVegSnowFreeDays # (*) snow free days for ArcVeg: is used to
define the starting day for arcVeg GDD and inundation calculation
$Writegrid                    # Writecode for this grid
$outpath//$ArcVegPlantGrowPeriods # number of growing periods for plants
depending on growing degree days (to be calculated within ArcVeg
$Writegrid                    # Writecode for this grid
$outpath//$ArcVegInundationScaling # Scaling factor for ArcVeg that is
calculated based on the ArcVegInundationAvg7DayMax grid within ArcVeg
$Writegrid                    # Writecode for this grid
$outpath//$ArcVegNmineralizationStack # AVNMIN Nitrogen Mineralization rate for
the actual year, separate for each plant grow period
$Writestack                    # Writecode for this stack
$outpath//$ArcVegNmineralizationTotal # AVNMINTOT total nitrogen mineralization
for the actual year (sum of all plant grow periods values from the stack above)
$Writegrid                    # Writecode for this grid
$outpath//$ArcVegSoilOrganicNitrogen # (*)AVSON total soil organic nitrogen
(needs to be read in as an initial grid with a default vaue of e.g. 1200 g/m^2)
$Writegrid                    # Writecode for this grid

→ (*) this stack must be created separately and it is read in when using
readgrids = 1, so these grids must be present in the StateIni folder from the
old run
```

Step 2: Adding a series of standard grids

A series of additional grids must be defined in the standardgrids section. These grids are variable over time, so for the real model runs, they must have a writecode of 3 (or any other code that writes the last grid of the model run in order to re-use these grids in subsequent model runs). At the end of the sample section, a short comment describes how to create these grids (content depends on the parameters in the multilayer landuse table).

ArcVeg_Fraction_1...5	Area fraction of the 5 landuse types that are used for ArcVeg in 1/1. The sum of all fractions must be 1.
ArcVeg_Biomass_1...5	Biomass for the 5 landuse types in g/m^2
ArcVeg_Retrans_1...5	re-translocated nitrogen for each plant type (in g/m^2)

```
# variables definitions
$set $IceWedgeUpperDepth    =  $grid//.iwdgup4
$set $IceWedgeLowerDepth    =  $grid//.iwdglo4
$set $ArcVeg_Fraction_1     =  $grid//.avfrac1
$set $ArcVeg_Fraction_2     =  $grid//.avfrac2
$set $ArcVeg_Fraction_3     =  $grid//.avfrac3
$set $ArcVeg_Fraction_4     =  $grid//.avfrac4
$set $ArcVeg_Fraction_5     =  $grid//.avfrac5
$set $ArcVeg_Biomass_1     =  $grid//.avbiom1
$set $ArcVeg_Biomass_2     =  $grid//.avbiom2
$set $ArcVeg_Biomass_3     =  $grid//.avbiom3
```

```
$set $ArcVeg_Biomass_4      =  $grid//.avbiom4
$set $ArcVeg_Biomass_5      =  $grid//.avbiom5
$set $ArcVeg_Retrans_1      =  $grid//.avretrans1
$set $ArcVeg_Retrans_2      =  $grid//.avretrans2
$set $ArcVeg_Retrans_3      =  $grid//.avretrans3
$set $ArcVeg_Retrans_4      =  $grid//.avretrans4
$set $ArcVeg_Retrans_5      =  $grid//.avretrans5

[standard_grids]
34                              # number of standard grids
…
…
…
$inpath_grid//$IceWedgeUpperDepth   IceWedgeUpperDepth      fillcode = 0
$inpath_grid//$IceWedgeLowerDepth   IceWedgeLowerDepth      fillcode = 0
$outpath//$ArcVeg_Fraction_1        ArcVeg_Fraction_1 fillcode = 0 writecode = 3
readcode = 1 # (*) for creation of this grid: fillcode = 1 defaultValue = -1
writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Fraction_2        ArcVeg_Fraction_2 fillcode = 0 writecode = 3
readcode = 1 # (*) for creation of this grid: fillcode = 1 defaultValue = -1
writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Fraction_3        ArcVeg_Fraction_3 fillcode = 0 writecode = 3
readcode = 1 # (*) for creation of this grid: fillcode = 1 defaultValue = -1
writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Fraction_4        ArcVeg_Fraction_4 fillcode = 0 writecode = 3
readcode = 1 # (*) for creation of this grid: fillcode = 1 defaultValue = -1
writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Fraction_5        ArcVeg_Fraction_5 fillcode = 0 writecode = 3
readcode = 1 # (*) for creation of this grid: fillcode = 1 defaultValue = -1
writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Biomass_1           ArcVeg_Biomass_1      fillcode = 0
writecode = 3 readcode = 1 # (*) for creation of this grid: fillcode = 1
defaultValue = -1 writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Biomass_2           ArcVeg_Biomass_2      fillcode = 0
writecode = 3 readcode = 1 # (*) for creation of this grid: fillcode = 1
defaultValue = -1 writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Biomass_3           ArcVeg_Biomass_3      fillcode = 0
writecode = 3 readcode = 1 # (*) for creation of this grid: fillcode = 1
defaultValue = -1 writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Biomass_4           ArcVeg_Biomass_4      fillcode = 0
writecode = 3 readcode = 1 # (*) for creation of this grid: fillcode = 1
defaultValue = -1 writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Biomass_5           ArcVeg_Biomass_5      fillcode = 0
writecode = 3 readcode = 1 # (*) for creation of this grid: fillcode = 1
defaultValue = -1 writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Retrans_1           ArcVeg_Retrans_1      fillcode = 0
writecode = 3 readcode = 1 # (*) for creation of this grid: fillcode = 1
defaultValue = -1 writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Retrans_2           ArcVeg_Retrans_2      fillcode = 0
writecode = 3 readcode = 1 # (*) for creation of this grid: fillcode = 1
defaultValue = -1 writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Retrans_3           ArcVeg_Retrans_3      fillcode = 0
writecode = 3 readcode = 1 # (*) for creation of this grid: fillcode = 1
defaultValue = -1 writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Retrans_4           ArcVeg_Retrans_4      fillcode = 0
writecode = 3 readcode = 1 # (*) for creation of this grid: fillcode = 1
defaultValue = -1 writecode = 8 readcode = 0 ;
$outpath//$ArcVeg_Retrans_5           ArcVeg_Retrans_5      fillcode = 0
writecode = 3 readcode = 1 # (*) for creation of this grid: fillcode = 1
defaultValue = -1 writecode = 8 readcode = 0 ;

# (*) → for creation of this grid: fillcode = 1 defaultValue = -1 writecode = 8
readcode = 0 ; → that means, that all these grids must be provided with a
proper initialisation which can be done when running the model with
```

```
$readgrids = 1 for a single timestep and using the parameters after the comment
instead of the bold printed, active parameters for that initialization. All
grids will be written with an extension "filled" and must be renamed to the
correct names after that initial run. Then, the correct parameters must be set
and the model can be run with global $readgrids = 1.
```

Step 3: Creating the ExcessIce stack

The ExcessIce-Stack stores ice that can be molten and is then assumed to leave the soil as runoff (interflow or direct flow). The assumption is, that the soil contains not only the usual fractions of soil particles, water and air, but another fraction of pure ice, that has the same volume as all other fractions together, i.e. 50% of the soil is ice, the rest is "normal" soil. So, if the ice is completely molten, the remaining soil will have the same thickness as the surrounding cells. However, since the initialization of the soil layers started at the normal (flat) surface, the soil column reaches deeper at the beginning. After the ice wedge is degraded, that base will still remain at the same level, so the soil sinks in at the surface and creates channels (polygons), where there were ice wedges before.

The excess ice stack must be initialized by using two additional grids, that will only be used for initialization (cold start):

- Ice wedge upper limit – this grids defines a depth in m below the surface, where the ice wedge starts (when traveling down the soil cloumn)
- ice wedge lower limit - this grid defines a soil depth in m below the surface, where the ice wedge ends.

The creation of these grids is probably the most challenging part of the data preprocessing. There are methods available to linearize polygon structures and get the ice wedge locations from there. However, the depth of the ice wedge is not easy to estimate. If there are ground measurements, those values could be taken for all the ice wedges of a smaller region. For this manual, the IceWedgeUpperLimit was choosen to be 0.1 m below soil surface and the IceWedgeLowerLimit was choosen to be equal to the elevation above sea level (which is between 4 and 5 m here and is totally arbitrary.

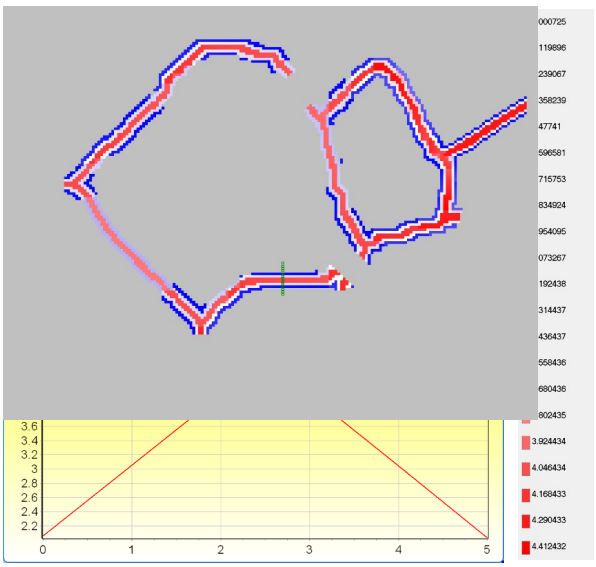


figure 2.29.2.1 IceWedgeLowerLimit (iwdglo) for the test basin (depth in cross section is between 2 and 4 m below surface)

Figure 2.29.2.1 shows the ice wedge lower limits, the cross section (green marked cells in the southernmost part) show the depth of 2 to 4 meters.

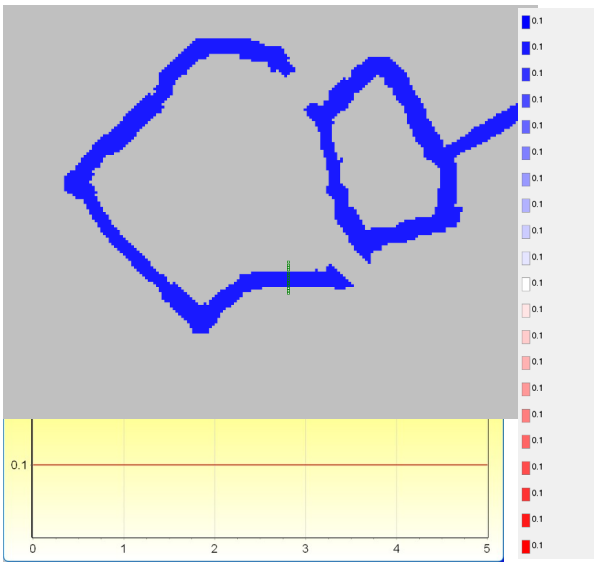


figure 2.29.2.2 IceWedgeUpperLimit (iwdgup) for the test basin (depth in cross section is at 0.1 m below surface for all calls on an ice wedge)

Figure 2.29.2.2 shows the according ice wedge upper limit, which is 0.1 m below soil surface here, i.e. the third soil layer is the first layer with ice.

When using the two grids iwdgup and iwdglo for initialization of the ExcessIce stack, the algorithm additionally adds some excess ice for all cells that are neighboring cells of the ice wedge, but only for half the thickness that those cells have (to imitate the wedge-shape a bit better when only a simple line is available from any automatic procedure of ice wedge localization).

The initialization of the ExcessIce stack is done when WaSiM is started in cold start mode with readgrids = 0. WaSiM looks for the grids containing the upper and lower ice wedge limits, creates the excess ice, and moves the soil layers accordingly downwards in order to give the excess ice the required space.

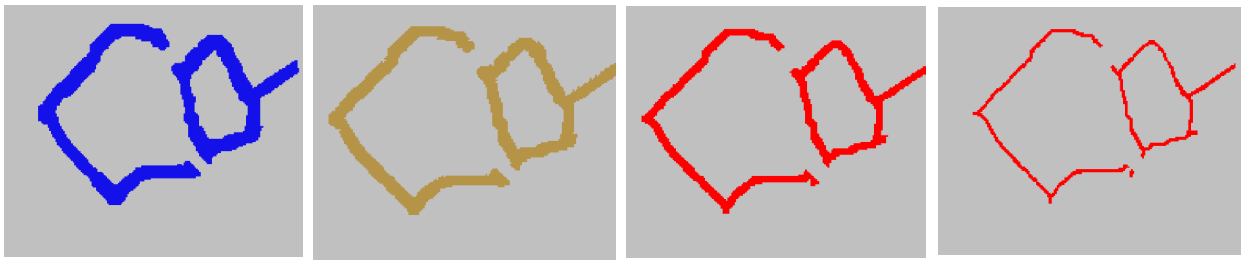


figure 2.29.2.3: Excess ice stack: Level 3, 9, 14 and 17 with 45.9, 41.8, 183.6 and 183.6 mm WE

Figure 2.29.2.3 shows some layers from the resulting excess ice stack, showing that the depth of the ice wedge is defined by the ice wedge lower limit. Also, the reghtmost grid shows that there are also additional cells on each side of the wedge added by the algorithm (i.e. these cells are not marked in neither the iwdgup nor iwdglo grids as ice wedge cells.

The ExcecssIce stack must be provided in the StateIni folder when readgrids is set to 1, so the initialization is not done for subsequent runs. Accordingly, the resulting stack of a model run stores the changed values of excess ice for each soil layer and must be provided in the statini folder for

any subsequent model run.

Step 4: DEM will be changed during the model run – procedure to use the correct one for subsequent runs

The resulting stack of the geodetic heights (hgeo) and the DEM itself will be stored after the model run. It is important to not only read in the hgeo stack for the next model run but also the changed DEM (which is stored under the same name but with an extended extension of "_out"). That grid should be renamed to the input name of the DEM grid for any subsequent runs. The hgeo stack is NOT initialized for every model run anymore. It's up to the modeler to keep the DEM grid and the hgeo stack in sync.

Step 5: Changes to the MultiLayer-LandUse-Table

compared to a normal application without ArcVeg integration, there are some more parameters to be set in the mult layer landuse table. Here is an example of what it looks like:

```
[multilayer_landuse]
2 # count of multilayer landuses
1  channel { Landuse_Layers = 1, 2, 3, 4, 5; ArcVeg_Initial_Fractions = 0.40,
0.30, 0.20, 0.05, 0.05; ArcVeg_Initial_Biomass = 500, 200, 180, 600, 400;
ArcVeg_Initial_Retrans = 0.30, 0.20, 0.10, 0.10, 0.00; k_extinct = 3.0;
LAI_scale = 1; }
2  polygon { Landuse_Layers = 1, 2, 3, 4, 5; ArcVeg_Initial_Fractions = 0.20,
0.10, 0.30, 0.30, 0.10; ArcVeg_Initial_Biomass = 300, 100, 200, 350, 250;
ArcVeg_Initial_Retrans = 0.35, 0.25, 0.15, 0.10, 0.05; k_extinct = 3.0;
LAI_scale = 1; }
```

As the parameter names already suggest, they are used for initzialization only.

ArcVeg_Initial_Fractions: The relative area of the grid cell that is used by this vegetation typevegetation type (the land use as defined in the land use table) . The sum must be equal to 1.0. Actual settings are for moss (0.4), tussoks (0.3), aquatic sedges (0.2), evergreen shrubs (0.05) and decennial shrubs (0.05) in arctic regions. **Note: actually (Release 11.00.01) the fraction is not variable, i.e. a changing biomass doen't lead to a change in the fraction of the vegetation types**

ArcVeg_Initial_Biomass: The biomass for each vegetation type (=land use as defined in the land use table) in g/m^2. Actual settings are for moss (300), tussoks (100), aquatic sedges (200), evergreen shrubs (350) and decennial shrubs (250) in arctic regions.

ArcVeg_Initial_Retrans: The retranslocated nitrogen in g/m^2 for each vegetation type according to the landuse layers. Actual settings are for moss (0.35), tussoks (0.25), aquatic sedges (0.15), evergreen shrubs (0.1) and decennial shrubs (0.05) in arctic regions.

When WaSiM starts in cold start mode with global readgrids parameter set to 0 and the standard grids for these entities set to the follwong paraeters:
fillcode = 1 defaultValue = -1 writecode = 8 readcode = 0 ;,
then after a short model run of e.g. 1 day the initial grids will be saved to disk with the extended extension *_filled. These grids should then be renamed to the correct name, placed in the stateini folder and then the normal run with readgrids = 1 can be started.

Step 6: Additional Parameters to the Landuse table

There are several parameters that must be added to the land use table for each plant type that is part

of the ArcVeg model. The follwing section shows a complete land use table entry for moss. :

```
[landuse_table]
5                    # number of following land use codes, per row one use
1 MOSS      {method    = VariableDayCount;
  ArcVegInFactFull = 1.00; # MOSS = 1.00, TUSS = 0.000, AQ-SEDG = 1.000, EV-
SHRUB = 0.0, DEC-SHRUB = 0.0 # a factor describing the effect of full
inundation on plant growth ability on an ice wedge
  ArcVegInFactSome = 0.12; # MOSS = 0.12, TUSS = 1.000, AQ-SEDG = 1.000, EV-
SHRUB = 0.2, DEC-SHRUB = 0.3 # a factor describing the effect of partial
inundation on plant growth ability for partial inundation on an ice wedge
  ArcVegInUDIceWdg = 1.00; # MOSS = 1.00, TUSS = 0.864, AQ-SEDG = 0.864, EV-
SHRUB = 0.7, DEC-SHRUB = 0.4
  ArcVegBNratio    = 125;   # MOSS = 125,  TUSS = 85, AQ-SEDG = 85, EV-SHRUB =
100, DEC-SHRUB = 100
  ArcVegSenesce    = 0.1;   # MOSS = 0.1,  TUSS = 0.3, AQ-SEDG = 0.25, EV-SHRUB
= 0.15, DEC-SHRUB = 0.15
  ArcVegRetran     = 0.00;  # MOSS = 1913   TUSS = 0.72     AQ-SEDG = 0.76
EV-SHRUB = 0.53    DEC-SHRUB = 0.55
  ArcVegFoliarProp = 1.00;  # MOSS = 1.00   TUSS = 0.20     AQ-SEDG = 0.20
EV-SHRUB = 0.19    DEC-SHRUB = 0.10
  ArcVegPropN      = 0.0081;# MOSS = 0.0081 TUSS = 0.0177   AQ-SEDG = 0.0188 EV-
SHRUB = 0.0108  DEC-SHRUB = 0.0187
  ArcVegAGProp     = 1.00;  # MOSS = 1.00   TUSS = 0.20     AQ-SEDG = 0.20
EV-SHRUB = 0.35    DEC-SHRUB = 0.25
  ArcVegMaxBio     = 1913;  # MOSS = 0.00   TUSS = 337      AQ-SEDG = 268
EV-SHRUB = 963        DEC-SHRUB = 963
  # climate zone     1        2     3     4     5     6     7
  ArcVegSeedMaxProb= 0.10     0.10  0.30  0.30  0.50  1.00  1.00 # maximum
probability in each year that a seedling will establish with low biomass (0.1
g/m2) for all seven climate zones (7 values are read in)
  # Plant Growing Period   1          2          3          4          5
  ArcVegUptakeEfficiency = 0.0075   0.0050      0.0050      0.0050      0.0050
  SoilTillage      = 9 10; # optional set of 1..n Julian days, depicting days
with soil tillage. Important for silting up model
  RootDistr        = 1;  # parameter for root density distribution
  TReduWet         = 1;  # relative Theta value for beginning water stress
(under wet conditions -> set >= 1 for crop which doesn't depend on an aeral
zone
  LimitReduWet     = 0.5; # minimum relative reduction factor of real
transpiration when water content reaches saturation. The reduction factor value
will go down
  HReduDry         = 10; # hydraulic head (suction) for beginning dryness
stress (for water content resulting in higher suctions, ETR will be reduced
down to 0 at
  IntercepCap      = 0.16; # optional: specific thickness of the water layer on
the leafes in mm. if omitted here, the dedfault parameter from
interception_model i
  JulDays            =  150   164   174   175   176   177   178   180   181
      182   183   184   185   194   195   196   204   211   212   214   215
      216   217   224   234   235   236   237   243   244   245   246   247
      248   254   264   266   267   268   284   320;  #     Julian      days
      for   all   following   rows.Eachparam
  Albedo             =  0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15
      0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15
      0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15
      0.15  0.15  0.15  0.15  0.15  0.15  0.15  0.15; #     Albedo (snow free)
  rsc          =  10    10    10    10    10    10    10    10    10    10    10
      10    10    10    10    10    10    10    10    10    10    10    10
      10    10    10    10    10    10    10    10    10    10    10    10
      10    10    10    10;   #     leaf surface resistance in s/m
  rs_interception =  1     1     1     1     1     1     1     1     1     1
      1     1     1     1     1     1     1     1     1     1     1     1
```

```
     1     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1;    #     INTERCEPTION      surface
     resistance  in    s/m
  rs_evaporation   =    40    40    40    40    40    40    40    40    40    40
     40    40    40    40    40    40    40    40    40    40    40    40
     40    40    40    40    40    40    40    40    40    40    40    40
     40    40    40    40    40;   #     SOIL surface
  LAI        =     0     0.1   0.1   0.1   0.11  0.12  0.13  0.14  0.15  0.16
     0.16  0.18  0.19  0.32  0.33  0.34  0.43  0.4   0.48  0.5   0.5   0.51
     0.51  0.53  0.53  0.53  0.52  0.52  0.49  0.49  0.48  0.47  0.47  0.46
     0.41  0.3   0.27  0.26  0.24  0.12  0;    #     Leaf Area Index (1/1)
  Z0         =     0.03  0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11
     0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11
     0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.11
     0.11  0.11  0.11  0.11  0.11  0.03  0.03; #     Roughness length in m
  VCF        =     0     0     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     0     1
     1     1     0     0;    #     Vegetation cover fraction (1/1)
  RootDepth  =     0     0     0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1
     0.1   0.1   0.1   0.1   0.11  0.12  0.15  0.17  0.18  0.2   0.2   0.2
     0.2   0.2   0.2   0.2   0.2   0.2   0.2   0.2   0.2   0.2   0.2   0.2
     0.2   0.2   0.2   0.2   0.2   0.2   0;    #     Root depth in m
AltDep       =     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0;    #     elevation depoendency of the Julian Day
(e.g. if the vegatation should start 3 weeks later when going up for 1000m, the
value may be set to 0.021 -> 1000m * 0.021 = 21 days)
  n-factor   =     1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
     1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
     1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
     1.0   1.0   1.0   1.0   1.0   0.1   0.3; # Factor applied to the air
temperature to estimate the ground surface temperature as boundary condition
for the soil temperature regime. Values are interpolated and not altitude
corrected.
# now some parameters for surface routing will follow. They are optional, the
default value is
# given at the start of the comment.
  SR_cultivation       = other; # default: other; type of cultivating the
surface; possible values: conserving (c), nonconserving (nc), other (o),
abbrev. used in the following rows are for convenience only
  (nc) SR_StemDiameter = 0.01 0.02 0.03 0.04 0.05 0.05 0.08 0.08 0.05 0.03 0.02
0.02; # default: 0.01 surface routing: diameter of the stalks or effective
diameter of crop/tree stems (perpendicular to the streamlines); used for
nonconserving land use (using ploughs)
  (c) SR_ResidCovrg    = 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ; #
default: 0.01 fraction of soil covered by vegetation, stones, and residuals
(mulch); used for conserving land use only
  (nc) SR_CropDistX    = 0.3; # default: 0.3 distance of crops in flow
direction; used for nonconserving land use (using ploughs) only
  (nc) SR_CropDistY    = 0.4; # default: 0.3 distance of crops perpendicular to
flow direction; used for nonconserving land use (using ploughs) only
  (nc/o) SR_K_ST       = 15; # default: 10 Manning-Strickler-roughness for the
soil surface in (m^1/3)/s; used for nonconserving and other landuse only
  (nc) SR_C_WR         = 1.5; #default: 1.5 form drag coefficient for a
circular cylinder in a group; used for nonconserving landuse only
  (c) SR_rough_coeff1  = 0.00496 0.892 0.0311; # default: 0.0198 0.712 0.142;
coefficients for k_st-calculation for conserving landuse, Reynolds number <
20000
  (c) SR_rough_coeff2  = 0.00519 1.200 0.177 ; # default: 0.0198 0.712 0.142;
coefficients for k_st-calculation for conserving landuse, Reynolds number >=
20000
}
```

Here are some basic parameter descriptions:

ArcVegInFactFull	Parameter for calculating an internal scaling factor for N-uptake efficiency depending on the location and inundation duration. Here for full inundation of > 300mm for at least 7 days
ArcVegInFactSome	Parameter for calculating an internal scaling factor for N-uptake efficiency depending on the location and inundation duration. Here for inundation of >0 ... 300mm for at least 7 days for non-ice-wedge cells
ArcVegInUDIceWdg	Parameter for calculating an internal scaling factor for N-uptake efficiency depending on the location and inundation duration. Here for inundation of >0 ... 300mm for at least 7 days for ice-wedge cells – the effective parameter is linearly scaled between ArcVegInFactSome (~300 mm) and ArcVegInUDIceWdg (~0 mm) depending on the max-7-day-inundation.
ArcVebBNRatio	Biomass to Nitrogen ratio (e.g. 125: each gramm nitrogen leads to 125 gramm of biomass)
ArcVegSenesce	fraction of the biomass that senesces each year
ArcVegRetran	re-translocated nitrogen (i.e. the plants can transfer that amount to the root zone before senescing)
ArcVegFoliarProp	proportion of biomass in the foliages relative to the total biomass
ArcVegPropN	proportion of nitrogen in the fiolages biomass
ArcVegAGProp	retranslocation proportion
ArcVegMaxBio	maximum biomass of a plant type in g/m^2
ArcVegSeedMaxProb	maximum probability for seeding, depending on the climate zone (0...1), one value per climate zone
ArcVegUptakeEfficiency	efficiency of nitrogen uptake for each plant growth period

Step 7: running the model with ArcVeg Modules

The following steps must be followed to ensure the ice wedge degradation and ArcVeg model to run successfully:

- run wasim with heat transfer model turned on, layered snow model turned on and without arcvegintegration in spinup mode to create a usefull soil temperature
- make all the additional entries needed in the control file (standard grids, arcVegIntegration section, multilayer_landuse table, landuse-table etc.)
- create ice wedge upper limit and ice wedge lower limit grids (manually or by any smart algorithm)
- set readgrids=0, set
  `fillcode = 1 defaultValue = -1 writecode = 8 readcode = 0 ;`
  for ll grids in the standard grids section that must be initialized first (av_biomass, AV_fraction and av_retrans)
- switch the spinup mode off
- run the model for a single day or time step-by-step. It will create the arcveg_biomass, arcveg_fractions and arcveg_retrans grids as well as the ExcessIce stack.
- rename all the created biomass, fraction and retrans grids to their correct name (i.e. remove the _filled-extension from the file extension) These grids will be located in the stateini folder
- the excessice stack also belongs into the stateini folder.
- change the parameters for these standard grids to the productive values
  `fillcode = 0 writecode = 3 readcode = 1`

- copy all storage grids in the stateini folder (bold: new grids and stacks for arcveg)
  (**25lcpavbiom1...5**; **25lcpavfrac1...5** **25lcpavretrans1...5**
  storage_richards.ftz_0...n albe25lcp.grd **avgdd25lcp**.grd
  avin7dmax25lcp.grd **avindavg25lcp**.grd **avinhlp25lcp**.grd
  avsnowfree25lcp.grd **avson25lcp**.grd **avtdavg25lcp**.grd
  denssnow_stack25lcp.grd **exice25lcp.grd** gwh125lcp.grd
  gwh125lcp_old.grd gwlv25lcp.grd gwst25lcp.grd
  gwth25lcp.grd hgeo25lcp.grd hgeosnow_stack25lcp.grd
  hhyd25lcp.grd hhydsnow_stack25lcp.grd qb__25lcp.grd
  sage25lcp.grd sday25lcp.grd sfstsu25lcp.grd
  si__25lcp.grd si__25lcp_Layer_1.grd si__25lcp_Layer_2.grd
  si__25lcp_Layer_3.grd si__25lcp_Layer_4.grd si__25lcp_Layer_5.grd
  sliq25lcp.grd snow25lcp.grd snowtemp25lcp.grd
  ssno25lcp.grd ssrftemp25lcp.grd ssto25lcp.grd
  swel_stack25lcp.grd swes_stack25lcp.grd tempersnow_stack25lcp.grd
  teth25lcp.grd thetasnow_stack25lcp.grd thicksnow_stack25lcp.grd
  tsoil25lcp.grd upperIndexSnow25lcp.grd)
- make sure that the dem grid in the input/grids folder is really the correct one (the dem will change when arcveg and the ice wedge degradation is running)
- add a special_output for geodetic head:

```
outputfile { header = geodetic_heads:
   filename = $outpath//special_output_geodetic_heads.//$code//$year;
   entity {
      ID      = geodetic_heads;
      Symbol  = hgeo;
      Xcoords =  585746, 585746, 585746, 585746, 585746, 585746, 585746,
 585746, 585746, 585746, 585746, 585746, 585746, 585746, 585746, 585746,
 585746, 585746, 585746, 585746, 585746, 585746, 585746, 585746, 585746,
 585746, 585746, 585746, 585746, 585746, 585746, 585746, 585746, 585746,
 585746, 585746, 585746, 585746, 585746;
      Ycoords =
7910478,7910478,7910478,7910478,7910478,7910478,7910478,7910478,7910478,
7910478,7910478,7910478,7910478,7910478,7910478,7910478,7910478,7910478,
7910478,7910478,7910478,7910478,7910478,7910478,7910478,7910478,7910478,
7910478,7910478,7910478,7910478,7910478,7910478,7910478,7910478,7910478,
7910478,7910478,7910478;
      Levels  =0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,
22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38;
   }
}
```

- run the model for the first year
  - check the hgeo-stack and the hgeo special output if the soil layers thickness was changed as expected. You can also look at the average thawing depth (ts_avg) to see how deep the soil should have been thawed up)
- create the stateini folder for the next model year, copy all grids and stacks and storage-richards files there; rename the dem_out file in the input/grids folder to the original dem_name (thereby overwriting the original DEM, so please make a backup first)
- run the second year , again, rename the dem_out grid to the original DEM file name and copy all required grids and stacks etc. to the new state ini folder
- and so on…
- tip: dothis with a batch job… :-)

Sample output

The following figures show some typical grids and time series for a small test basin, modelled in 0.25 m resolution for an area of around 25 x 40 m. To get some visible changes, the arctic climate

was changed by +10°C, also an extra portion of rain was allowed for some days after the snow melt period in order to fill up the channels that were created by ice wedge degradation.

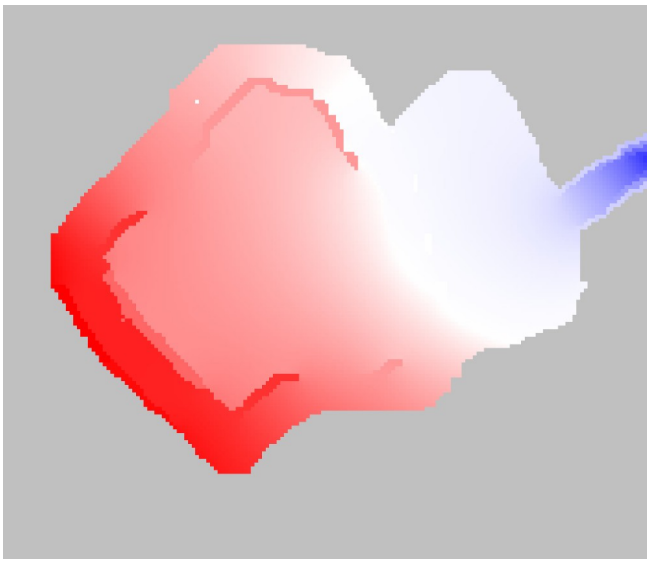


figure 2.29.2.4: original DEM (almost) without any channels

Figure 2.29.2.4 shows the original DEM with almost no variations in elevation. The ice wedge is located according to figures 2.29.2.1 (ice wedge upper limit for initialization), 2.29.2.2 (ice wedge lower limit for initialization), and 2.29.2.3 (excess ice stack). After one year of modeling with a temperature rise of +10°C and an extra portion of rainfall during 8 days (May 20th to 28th with 1.5625 mm/h = 300 mm in 8 days) the resulting DEM is shown in figure 2.29.2.5. The cross section shows a significant degradation where there are ice wedges and no changes where there are no ice wedges (as expected).

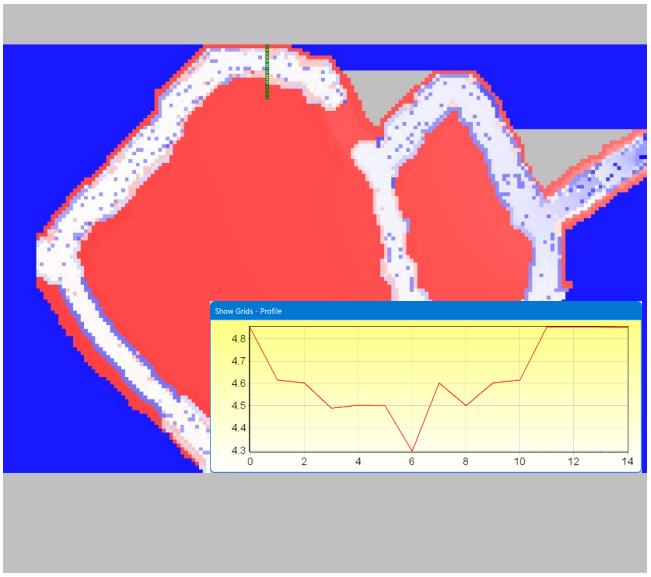


figure 2.29.2.5: DEM after year 1, the cross section shows the decreased soil surface over the degraded ice wedges

However, that channel was already filled up partially by heavily grown moss. The real degradation is shown in figure 2.29.2.6 at around the beginning of October for a single cell over an ice wedge.

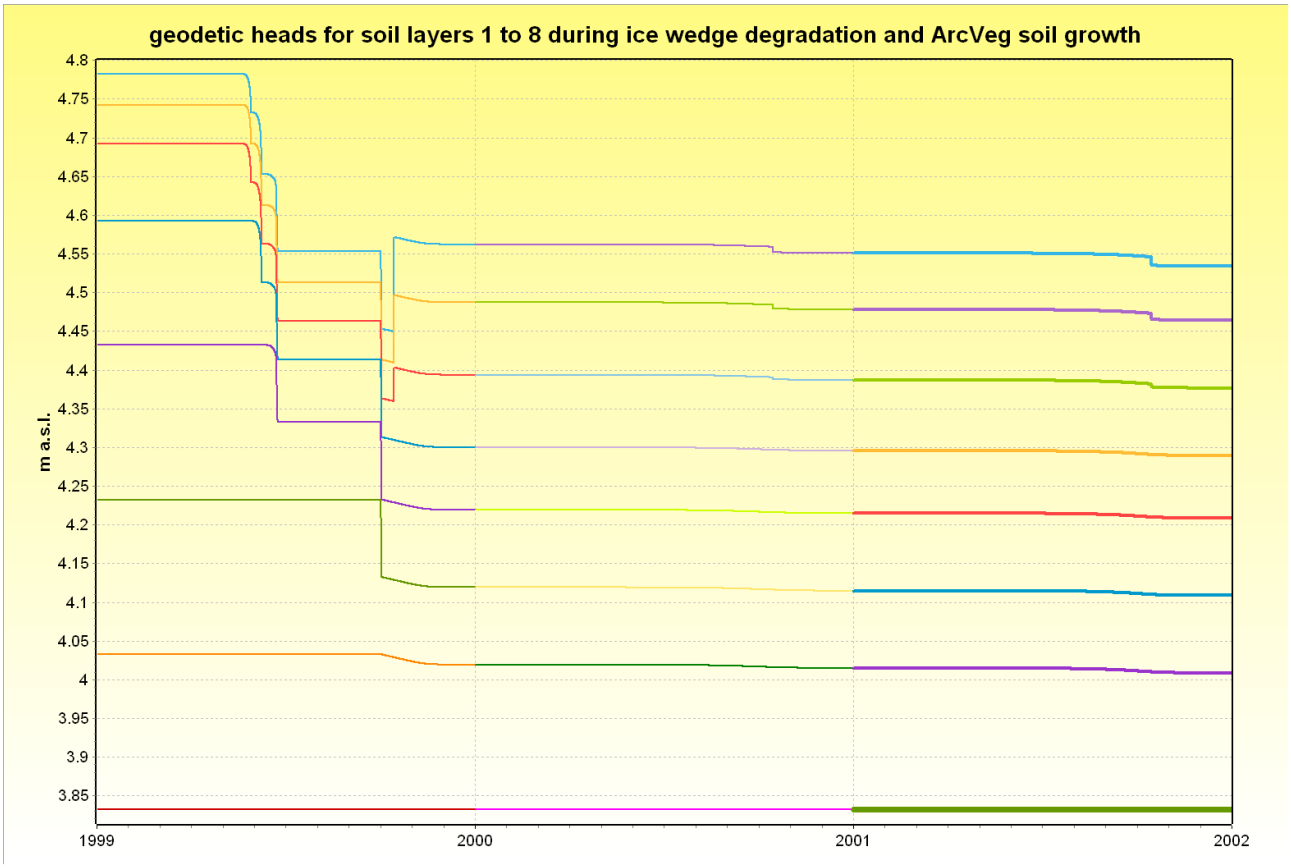


figure 2.29.2.6: time series for geodetic elevation of the uppermost 8 soil layers during ice wedge degradation (1999) and the following stabilazation phase (2000 and 2001)

On October 15th, 1999, the ArcVeg Module runs and the soil "growth" at this time by approximately 12 cm. This is of course not realistic, but here we used +10°C temperature rise as well as +300 mm rainfall and the Available nitrogen in the soil (soil organic nitrogen AVSON) was initialized by a quite high value, so ArcVeg resulted in a big biomass growth for moss, which is then translated into soil thickness change of 12 cm. That means that there is 12 x 100 g = 1200 g new moss biomass after the first year. This biomass degrades a little bit during the following years, so the soil gets a bit thinner again.

The shown examples only demonstrate the capabilities of the combined ice wedge degradation and the "Arctic tundra vegetation dynamics model ArcVeg". Further systematic applications will be done to verify the applicability to larger basins. This can probably only be done on larger super computers, since a) the spatial resolution must be very high, b) the temporal resolution for the surface runoff model must be very high, too (some seconds) and c) the vegetation development needs several decades to centennials to come into a stable state.

2.30 Automatic Soil Type Creation

Providing spatially distributed soil data in high resolution can be a challenge. Often, the soil types are only rough estimates, derived from coarser data sets. However, for some regions or countries, more detailed data are available. A data set provided for Switzerland in a 90x90 m resolution are the KOBO Maps (KOBO Hinweiskarten). They provide for large parts of Switzerland the fractions of sand, silt and clay in three different soil depths ranges: 0...30 cm, 30...60 cm and 60...120 cm.

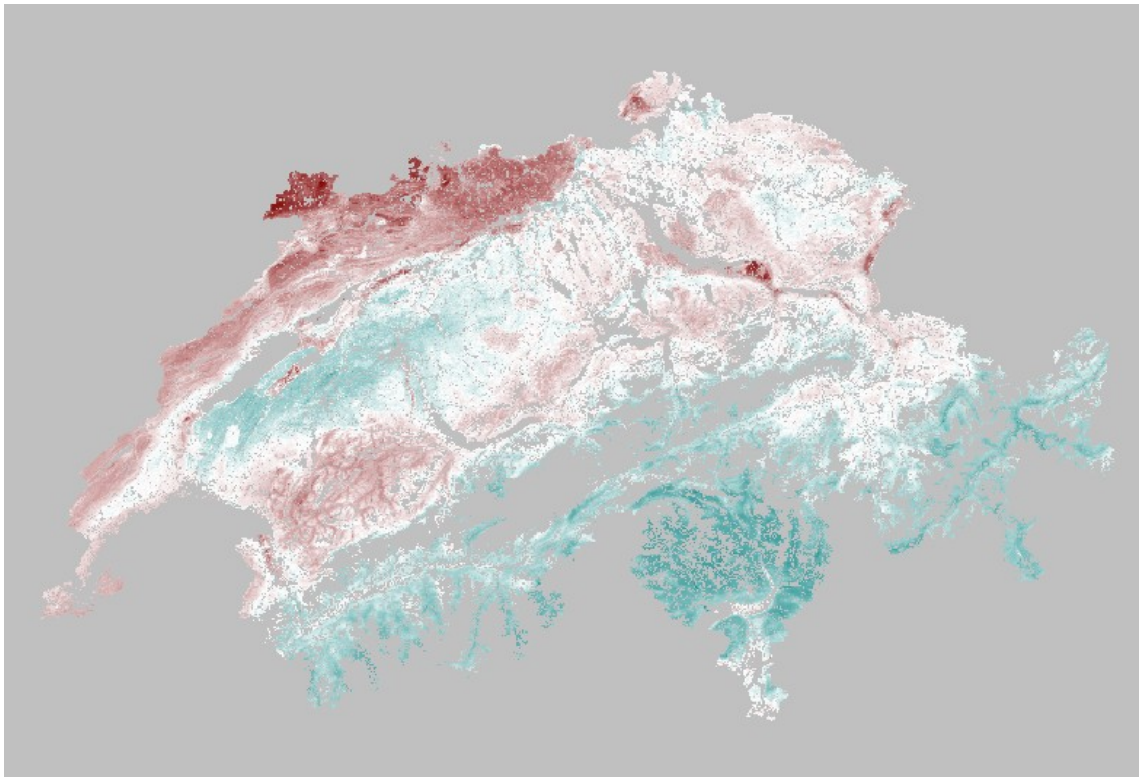


figure 2.30.1: sand fraction as a medium value for 0...120 depth, ranging from around 2% (brown) to 70% (blue)

Figure 2.30.1 shows an example for such a map. The shown values range from very low sand fractions of around 2 to 5 % in North-West part of Switzerland to quite high values of up to 75 % in the Southern regions. The shown map shows averaged data for 0...120 cm soil depth, already resampled to 500 m resolution.

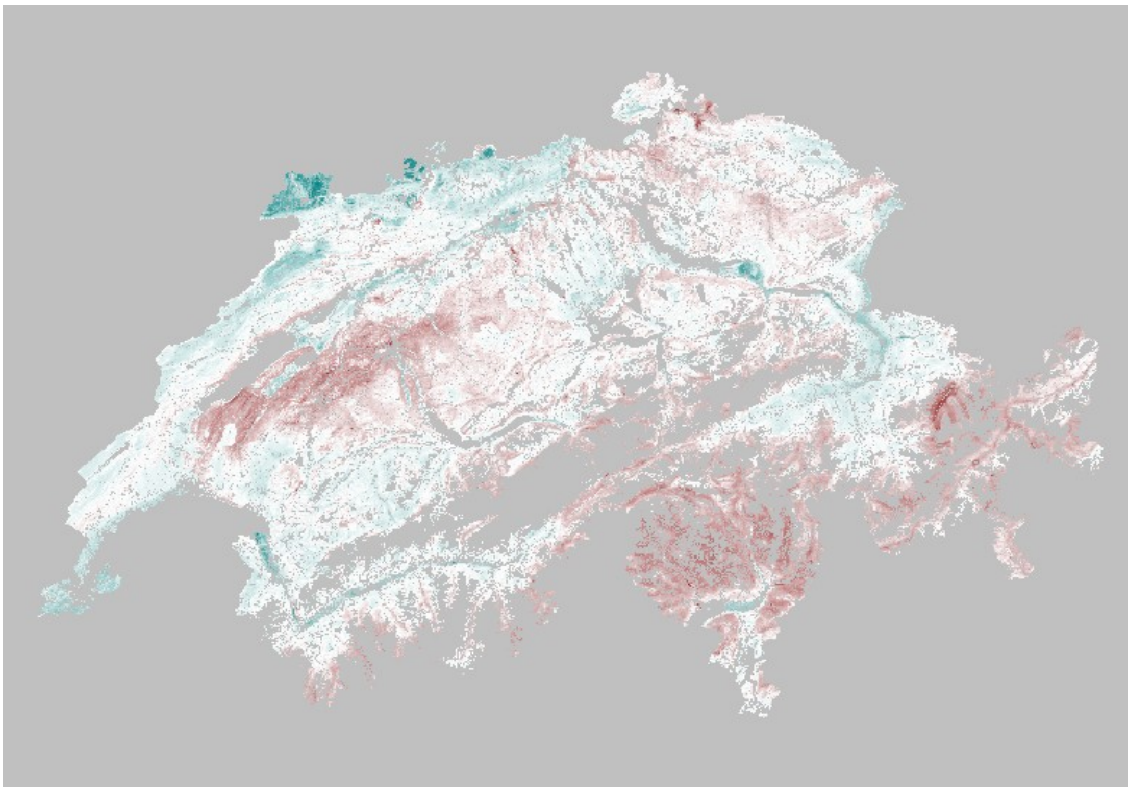


figure 2.30.2: silt fraction as a medium value for 0...120 depth, ranging from around 14% (brown) to 70% (blue)

Figure 2.30.2 shows the silt fraction for the same region, and finally figure 2.30.3 shows the clay fraction.

In all of these maps it can be seen that the coverage is not complete: All areas that are rock, all sealed areas (cities, traffic areas etc. and all lakes and rivers are not covered by this data set.

WaSiM offers a way to use this data to create soil types automatically. It is highly recommended to use a corresponding soil table, described later on in this chapter.

Basis for the soil type computation is the USDA soil texture triangle, see figure 2.30.4. If the control file contains the names of the clay, silt and sand fraction grids as standard grids, then WaSiM assumes to use them for automatic soil type creation. Note: A default soil type grid is neccessar anyway in order to provide soil types for uncovered areas and soil depth for all cells (which is taken from the soil table according to the default soil type at that same location).

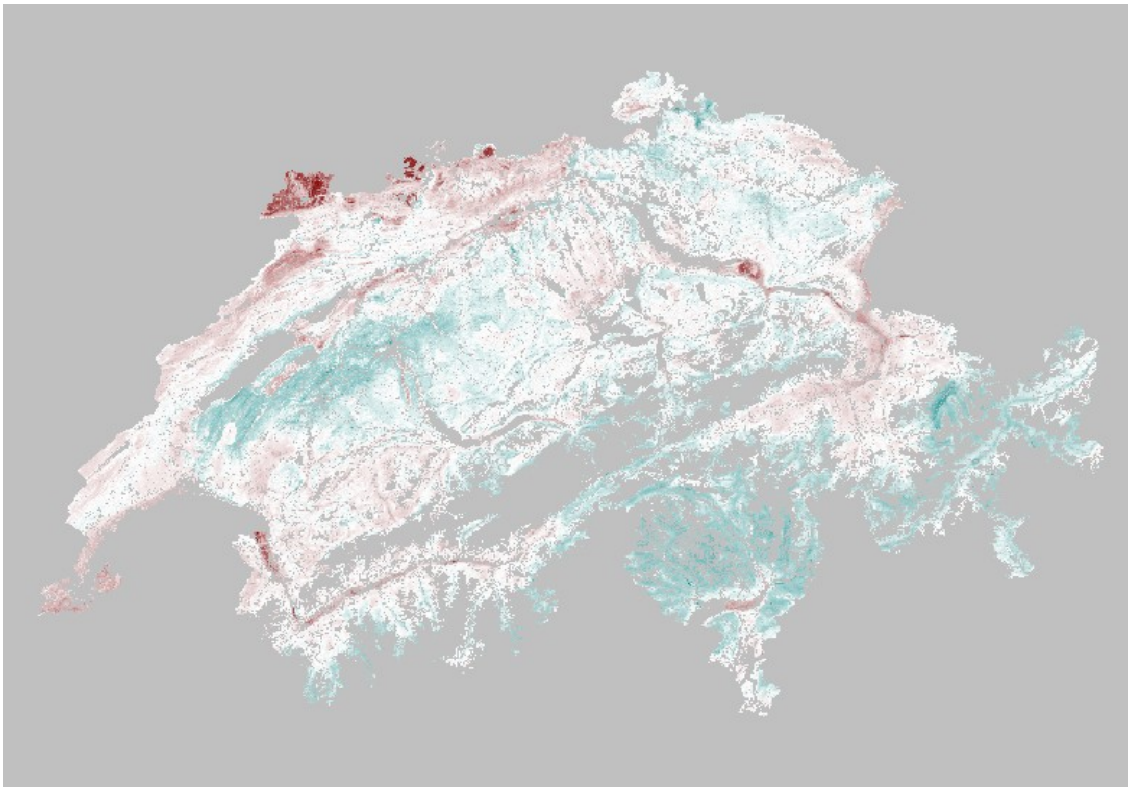


figure 2.30.3: clay fraction as a medium value for 0...120 depth, ranging from around 30% (brown) to 85% (blue)

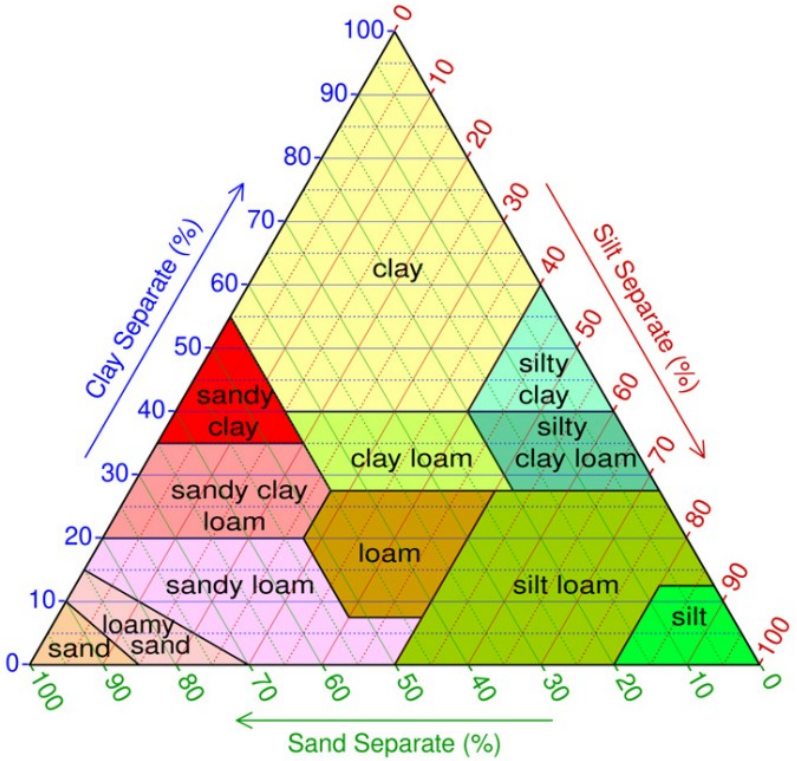


figure 2.30.4: USDA Soil texture triangle (taken from https://drainagematters.org/stormwater-glossary/usda-soil-texture-triangle.html, 2026-11-05)

Depending on the relative share of each of the trheee texture compartments, the place in the soil texture triangle is located by the algorithm:

Soil table ID	Soil type	total depth in m	Layers in horizons A-C	Layers on horizon D
2	Silty_Clay_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
22	Silty_Clay_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38
42	Silty_Clay_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
3	Clay_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
23	Clay_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38
42	Clay_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
4	silty_clay_loam_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
24	silty_clay_loam_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38
44	silty_clay_loam_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
5	clay_loam_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
25	clay_loam_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38
45	clay_loam_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
6	silt_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
26	silt_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38
46	silt_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
7	silty_loam_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
27	silty_loam_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38
47	silty_loam_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
8	sandy_clay_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
28	sandy_clay_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38
48	sandy_clay_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
9	loam_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
29	loam_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38

Soil table ID	Soil type	total depth in m	Layers in horizons A-C	Layers on horizon D
49	loam_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
10	sandy_clay_loam_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
30	sandy_clay_loam_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38
50	sandy_clay_loam_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
11	sandy_loam_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
31	sandy_loam_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38
51	sandy_loam_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
12	loamy_sand_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
32	loamy_sand_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38
52	loamy_sand_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
13	sand_shallow	2.5	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.13
33	sand_medium	5.0	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 0.38
53	sand_deep	16.2	3 x 0.1, 2 x 0.15, 4 x 0.15	10 x 1.5
14	Lake bottoms	10.5	9 x 0.1, 2 x 0.3, 9 x 1.0	-
15	Settlements	4.5	9 x 0.1, 3 x 0.2, 8 x 0.375	-
16	Rock	2.5	7 x 0.1, 3 x 0.1, 5 x 0.1	5 x 0.2
17	Bog, moor, swamp etc.	4.0	9 x 0.1, 3 x 0.2, 8 x 0.375	-

Note that each soil type has three incarnations: shallow (default depth 2.5 m), medium (default depth 5.0 m) and deep (default depth 16.2 m). If the soil depth should be changed globally, the parameter for thickness scaling can be used. However, this will also change the upper three horizons A to C, which are explicitely not valid below 1.2 m. So, when using this soil table together with the shown clay-, sand- and silt-fraction grids, it's better to scale the total thicknett of the soil by using the D horizon for scaling, e.g. by using three variables $LayerDepthDshallow, $LayerDepthDmedium and $LayerDepthDdeep with their values initially set to 0.13, 0.38 and 1.5 m, respectively.

When the algorithm computed the general soil type, e.g. sandy loam as code 11, it looks in the original SOIL grid for the code, uses this as index in the soil_table and takes the total thickness from that code as indicator for the incarnation to use: shallow soils are between not thicker than 1.2 x the default thickness of the shallow soil types in the table above (i.e. not deeper than 2.5*1.2 = 3.0 m). Medium soils are between 3 and 1.2*5=6 m thick, and deep soils are deeper than 6 meters. The example above of sandy loam would result in code 11, when the original soil table entry was less than 3 m, or in code 31 when the original soil table entries thickness was between 3 and 6 m

and 51 for all other thicknesses.

Once the new code is created, the value in the SOIL grid will be replaced by this new value. A grid with the name “ComputedSoils_0” will be written to the default output directory of the model run. Note that this grid does contain the compressed soil table range without unused indices, as it is used internally, so the soil type codes range from 1 to 40, not from 2 to 53 and they are sorted by occurrence in the soil table, i.e. the original soil codes 2, 23 and 43 will be codes 1, 2 and 3 in the exported grid.

3 Overview of required input data, parameters and data flow charts

3.1 Input data requirements for some model setups

WaSiM allows various model configurations depending on the aim of the application and on the amount and quality of input data. It is possible to combine various submodel components and to run the model in various spatial and temporal discretizations.

minimum requirements

The minimum meteorologic input data are temperature and precipitation – for flood forecasting without snowmelt, even temperature may be ignored. The modules for interception, snow, infiltration (model version 1) and discharge routing may be disabled during the model run. Also the irrigation and the substance transport can be disabled.. Thus, a minimal configuration for a meaningful water balance model would be a combination of data interpolation for precipitation and temperature, evaporation and runoff generation (soil model). If only a single flood event has to be modeled (using WaSiM version 1 with the TOPMODEL-approach), then even the evaporation module may be disabled and no temperature is needed as well. Then, only precipitation data and the digital elevation model, land use and soil type data has to be read in. This minimal configuration does not work if there is much snow, but to switch on the snow model does mean only that the temperature will be needed in addition to the precipitation. If the evaporation approaches after HAMON, WENDLING or HAUDE should be applied, the model can run only in a daily time step or longer.

requirements for an optimal simulation

For a long term simulation in a high temporal resolution all components which were described in the first part of this documentation should be enabled (except irrigation and substance transport, if this is not important and probably heat transfer and some special features). All of these components should be run in a sufficiently high spatial and temporal resolution. Compared to the minimum configuration there are considerably more input data required. Also the parameterization and the calibration is more complex. Table 3.1.1 contains a short overview for three different applications (only model version with Richards-approach is considered here).

Table 3.1.1: required submodels and model-input data for different model objectives

	flood simulation (only events)	**long term water balance simulation (daily time step)**	**continuous simulation including floods**
resolution			
• temporal	1 min … 1 h (max 1 d)	1 d	≤ 1 h (max. 1 d)
• spatial (as grid)	depending on the basin, e.g. 2…5 m for basins < 1 km ^{2} … 5 km for basins ≥ 10,000 km ^{2}		
sub-modules (components)			
• precipitation correction	**Required**	**Required**	**Required**
• RegionalSuperposition	not required (only for large basins with different precipitation regimes)	not required (only for large basins with different precipitation regimes)	not required (only for large basins with different precipitation regimes)
• input data interpolation	**Required (at least precipitation, temperature is recommended)**	**Required (at least temperature and precipitation;** radiation, wind speed, air humidity or vapor pressure and sunshine duration are recommended	**Required (at least temperature and precipitation;** radiation, wind speed, air humidity or vapor pressure and sunshine duration are recommended
• radiation	not required	Not required but	not required but

correction/temperature modification		recommended	recommended
• evaporation model	not required	**required** , Wendling, Hamon or Penman-Monteith	**required** , if possible after Penman-Monteith
• snow model	required only when snow accumulation or snow melt is important	**required**	**required**
• Glacier model	**Required**, (but only when glaciers are important)	**Required,** (but only when glaciers are important)	**Required,** (but only when glaciers are important)
• interception model	recommended	**Required**	**Required**
• infiltration model (only Topmodel version)	Highly recommended	not required	recommended
• soil model / unsaturated zone model (both model versions)	**required**	**Required**	**Required**
• SiltingUp (only for special small scale applications)	Not required	Not required	Not required
• Surface routing (only for small scale applications in flat terrain)	Not required	Not required	Not required
• groundwater model (Richartds version only)	not required	**Highly recommended**	**Highly recommended**
• Discharge routing model	**Required,** for large or subdivided basins	**Required,** for large or subdivided basins	**Required,** for large or subdivided basins
• Lake model	**Highly recommended,** but technically not required	**Highly recommended,** but technically not required	**Highly recommended,** but technically not required
• Heat transport	**Not required**	**Recommended** when soil freezing is important for runoff generation (mountainuous regions)	**Recommended** when soil freezing is important for runoff generation (mountainuous regions)
• Irrigation	Not required	**Recommended** if dealing with agricultural areas with irrigation effects	**Recommended** if dealing with agricultural areas with irrigation effects
• substance transport (only version 2)	not recommended for events (but possible)	if required	if required
spatial distributed data (grids)			
• digital elevation model	**Required**	**required**	**required**
• aspect (°)	not required	**required** (if required [1])	**required** (if required [1])
• slope angle (°)	not required	**required** (if required [1])	**required** (if required [1])
• topographic index $\ln(a \cdot (T_0 \cdot \tan\beta)^{-1})$	**required for model-version 1**	**required for model-version 1**	**required for model-version 1**
• firn and ice (glacier grid static glacier model)	**only required and allowed if glaciers are modeled and static method is choosen**		
• Glacier IDs	**Only required when using dynamic glacier model**		
• Glacierized fractions	**Only required when using dynamic glacier model**		
• zones for sub basins	**required** [2]	**required** [2]	**required** [2]
• land use	**required** [3]	**required** [3]	**required** [3]
• soil types	**required** [3]	**required** [3]	**required** [3]
• Flow time sums	**required**	Not required	**required**
• River depths	Not required	**Recommended** (when	**recommended** (when

		dealing with groundwater ↔ surface water interactions	dealing with groundwater ↔ surface water interactions
• River widths	Not required	**Recommended** (as above)	**Recommended** (as above)
• River links	Not required	**Recommended** (as above)	**Recommended** (as above)
• Colmation	Not required	**Recommended** (as above)	**Recommended** (as above)
• vertical hydraulic conductivity (K_s-value in $m \cdot s^{-1}$, version 1)	may be estimated from soil data, its also possible to read in separate grids with this data; only used in Topmodel version.		
• water content at field capacity [mm] (version 1)	may be estimated from soil data, its also possible to read in separate grids with this data; only used in Topmodel version		
groundwater model	The use of the groundwater module is recommended for low land catchments with huge sediment layers or if the groundwater is very important for the hydrologic problem. It can be disabled in mountainous regions. (only available in model version 2)		
• K_S-value in x-direction for each aquifer [m/s]	not required	recommended	recommended
• K_S-value in y-direction for each aquifer [m/s]	not required	recommended	recommended
• aquifer thickness for each aquifer [m]	not required	recommended	recommended
• storage coefficient for each aquifer [m^3/m]	not required	recommended	recommended
• leakage factor for each aquifer to the next lower aquifer [s^{-1}]	not required	recommended	recommended
• boundary condition h=const for each aquifer [m]	not required	recommended	recommended
• boundary condition q=const for each aquifer [m]	not required	recommended	recommended
meteorological data (station data)			
• precipitation (mm)	**required**	**required**	**required**
• air temperature (°C)	**required** [4]	**required**	**required**
• air humidity (1/1 or mbar) [5]	not required	not required	**required** [5]
• global radiation ($Wh \cdot m^{-2}$)	not required	**required** at ET after WENDLING [6]	**required** [6]
• rel. sunshine duration (1/1)	not required	**required** at ET after WENDLING [6]	**required** [6]
• wind speed ($m \cdot s^{-1}$)	not required	not required	**required**
• hydrologic data (discharge data from gauging stations at rivers)			
• subbasin or basin runoff as specific runoffs [mm]	recommended or **required** [7]	recommended or **required** [7]	recommended or **required** [7]
other data			
• land use parameters (table)	**required**	**required**	**required**
• soil parameters (table)	**required**	**required**	**required**
• table with tracer properties	if required (not recommended), only in WaSiM-Version 2 with	if required, only in WaSiM-Version 2 with RICHARDS -equation	if required, only in WaSiM-Version 2 with RICHARDS -equation

		RICHARDS -equation		
•	Irrigation codes	Not required	**Recommended** when dealing with irrigation	**Recommended** when dealing with irrigation
•	Lake codes	**Recommended** when dealing with irrigation	**Recommended** when dealing with irrigation	**Recommended** when dealing with irrigation

[1] This data sets are required if using the radiation correction or temperature modification. They can also be derived from the elevation model from within the model

[2] A zone grid can contain zones determined by any specific property like subbasins, altitude belts, aspects a.s.o. The minimum would be a grid with unique values (i.e. all cells would have the same value), then there is only one zone, else: per zone a integer code >0;
It's possible to use two zone grids, one for the submodels of interpolation, interception, evaporation and snow and another for the soil- and groundwater model, irrigation and routing. For the first one the properties may be selected from the upper list, for the second one the zones must be hydrologic subbasins or at least closed regions with hydraulic interconnections by flow-channels because for these submodels a hydrologic subdivision is required.

[3] The minimum would be a unique code for all grid cells, this would mean a constant soil or land use all over the basin. There can be as many soil types and land uses as required. For each code in these grids there must be a matching entry in the soil table and in the multilayer_landuse table, respectively, in the control file with the proper parameters

[4] required only, if the snow model is enabled

[5] only one of the variables vapor pressure and relative air humidity is really required

[6] sunshine duration and global radiation may be derived from one another (see chapter 1.4.1), so one of the variables would be sufficient (with advantage the global radiation), but for a better estimation of the long wave radiation balance observed sunshine duration would be better

[7] To allow a calibration at observed runoff data there should be runoff data for at least a subset of the model time. The observed runoff data are needed only for computing the model efficiencies

3.2 Overview of the most important model parameters

Table 3.2.1 gives an overview about the most important free model parameters. Many of the parameters can be taken from the literature. A sensitivity analysis of the most important parameters is contained in chapter 4.15 of this documentation. The free model parameters are those which have only a low physical background or which can not be obtained easily by measurements. Examples for free parameters are the weighting p of the distances for the inverse distance interpolation, the drainage density for the unsaturated zone model and also some soil parameters (even if measurable, the effective value on a large grid cell may have to be subject to a calibration procedure). But in general, parameters which describe typical properties of soils and vegetation should not be calibrated, like the parameters describing the $k(\Theta)$-and $\psi(\Theta)$-functions or the root depth, leaf area index and resistances of given crop types. These parameters should be seen as "derived internal properties" of an observable parameter like soil type or crop type (however, feel free to calilbrate every parameter if it turns out to be the only way to get good results – as long as the physical meaning is not perverted.

Table 3.2.1: Parameters of the submodels

submodel	parameter	unit	comment
precipitation correction	a_l	-	from the literature, calibration if required
	b_l	mm/(m · s^{-1})	
	a_s	-	
	b_s	mm/(m · s^{-1})	
	$T_{R/S}$	°C	
interpolation of meteorological input data	d_{max}	km	max. distance to the next metorol. station (e.g. 100 km)
	p	-	power of the distance weighting for IDW-interpolation (literature: 2 … 3)
	igo, igu	m	upper and lower inversion layers, dependent on the basin (Thur basin: 800

submodel	parameter	unit	comment
			resp. 1400 m a.s.l.)
modification of temperature	c_T	K	scaling factor after, e.g. 5 K, calibration recommended
land use table for evaporation and interception model and ETR-extraction from the soil (for each crop type an entry in the land use table, parameter should *not* be calibrated)	r_c	$s \cdot m^{-1}$	resistances: 12 monthly values, literature,
	LAI	-	leaf area index: literature, measurements
	v (= 1 -A)	-	vegetation coverage degree: literature
	z_0	m	effective crop height: literature, measure
	α	-	Albedo: Literature, measure
	z_w	m	root depth: literature, measure
	$d_{1,400}$ - $d_{4,400}$	Jul. Day	phenologic course: literature, observations
	in addition for		
	RICHARDS :	-	root density distribution, literature, measurements
	ρ	m	minimum suction for reducing *ETR* compared to *ETP*
	ψ_g		
snow model	$T_{R/S}$	°C	transition temperature snow/rain: literature
	T_{trans}	K	
	T_0	°C	transition range snow/rain: literature
			threshold temperature melt, literature, measure, calibrate
	c_0	$mm \cdot °C^{-1} \cdot d^{-1}$	degree-day-factor: calibrate
glacier model (as an integral component of the snow model)	DDF_{ice}	$mm \cdot °C^{-1} \cdot d^{-1}$	degree-day-factor for ice (T-index-meth.)
	DDF_{firn}		degree-day-factor for firn
	DDF_{snow}		degree-day-factor for snow
	MF		melt-factor for ice/snow/firn (unique)
	α_{ice}	h	radiation coefficient for ice
	α_{firn}	h	radiation coefficient for firn
	α_{snow}	h	radiation coefficient for snow
	k_{ice}		recession parameter single linear stor. ice
	k_{firn}		recession parameter firn
	k_{snow}		recession parameter snow
interception model	h_{SH}	mm	maximum water layer thickness, calibr. v and LAI from land use table
infiltration model (topmodel only)	x_f	-	fraction of re-infiltrating water, calibrate, soil properties Ks and ψ_f from soil table
Silting up model	Rainpause	d	Minimum rainpause to separate two precipitation events
Surface routing	dT_{min}	s	Minimum sub time step
	dT_{max}	s	maximum sub time step
	f_c	-	Concentration factor
soil model in WaSiM-ETH version with TOPMODEL -approach	m	m	recession parameter, from observed hydrographs
	T_{korr}	-	transmissivity-correction: calibrate
	K_{korr}	-	K_s-correction: calibrate
	k_D	h	storage coefficient surface runoff: from a hydrograph analysis
	SH	mm	maximum content interflow storage: calibrate
	k_H	h	storage coefficient interflow: from a hydrograph analysis
	P_{grenz}	mm	max. precipitation intensity for matrix infiltration: calibrate, literature

submodel	parameter	unit	comment
	r_k	-	limitation of capillary rise: calibrate (0 … 1)
	QD_{Snow}	-	fraction of surface runoff on snow melt: calibrate (0 … 1)
soil model	k_D	h	storage coefficient surface runoff
after RICHARDS -equation	k_H	h	storage coefficient interflow (k_H here should consider only the storage effects of the surface channels like for surface runoff : recommended is $k_D = k_H$
	Q_0	mm/h	scaling factor for base flow
	k_B	m	storage coefficient for base flow (Q_0 and k_B are used only if the groundwater model is disabled)
	d_r	m^{-1}	drainage density for interflow: linear scaling of interflow ($d_r > 0$)
	QD_{Snow}	-	fraction of surface runoff on snow melt: calibrate (0 … 1)
groundwater model (Gauss-Seidel-algorithm)	n_{max}	-	maximum number of iterations per time step and layer
	ε_{iter}	m	maximum iteration error between two iteration steps (is multiplied with a factor of 50 at confined conditions)
	α_s	-	weighting of the head of the time step end (degree of implicity of the solution), 0 = fully explicit, 1=fully implicit
	SOR	-	successive over relaxation: factor for accelerate the convergence of the iteration. reasonable between 1 and 2, often 1.1..1.4; if SOR <0: automatic computation of SOR from the last three iteration steps
Soil table	TOPMODEL		
(for each soil type an entry	FC	Vol.-%	available field capacity, only TOPMODEL
exists in the soil table, these	mSB	Vol.-%	max. available soil water content
parameters should *not* be	ψ	mm	suction at the wetting front after (52)
calibrated , exclusion: c_k at	K_s	m/s	saturated hydraulic conductivity, measure, literature values.
larger time steps and k_{rec})	RICHARDS		
	Θ_{sat}	-	soil specific, measure or from the
	Θ_{wp}	-	literature
	K_s	m/s	sat. hydraul. conductivity, measure, liter.
	α	m^{-1}	van Genuchten parameter, literature
	n	-	van Genuchten parameter, literature
	k_{rec}	-	K_s -recession $K_s(t) = K_{s,t=0} \cdot (k_{rec})^t$, t : depth
	l_v	m	number of soil layers
	d_z	-	layer thickness
	c_k		limitation of the K_s -relation between two soil layers, useful for large time steps
discharge routing	B_h , B_v	m	width of mean channel and flood plains
	T_h	m	channel depth,
hydraulic parameters have to be given for each channel and for each external inflow;	M_h , M_v	$m^{1/3} \cdot s^{-1}$	roughness values for mean channel and flood plains, channel slope,
	I	$m \cdot m^{-1}$	channel length
	L	m	catchment area
	A_E	km^2	reservoir constants for storage effects for mean channel and flood plains
	k_h , k_v	h	threshold for extractions
threshold parameter for internal/external abstractions	Q_{min}	m^3/s	relative fraction of abstracted water on $Q - Q_{min}$
	Q_{rel}	m^3/s	

submodel	parameter	unit	comment
initial concentrations and reservoir contents for reservoirs		m^3/s	maximum capacity of the abstraction
	Q_{max}	m^3	reservoir content at model start
	V_0	kg/kg	initialization concentrations of all possible 9 tracers within the reservoirs at model start (also if no substance transport is considered, also in WaSiM-version 1)
	$C_{0,1} \ldots C_{0,9}$		
substance transport	$t_{½}$	d	half time (for radioactive tracers)
	c_{min}	kg/kg mg/l	minimum and maximum tracer concentrations (if useable, else -9999)
	c_{max}	kg/kg mg/l	
irrigation table	ψ_{dry}	m	threshold suction for starting irrigation
	ψ_{wet}	m	target suction value for stopping irrigation
Heat transfer	C_{dry}	J/(Kg*K)	Heat capacity of “dry” soil
	ρ_{dry}	Kg/m^3	Dry density of soil
	λ_{dry}	J/(m*s*K)	thermal conductivity for *dry” soil
	$k_{sat,f}$	m/s	hydraulic conductivity for froozen soil

4 Using WaSiM

The following sections give an overview on how to use WaSiM. It is written as a kind of step-by-step instruction in order to enable also beginners to use the model. If anything is missing (description of some important procedures needed to start the model etc.), please contact the auther.

4.1 Technical requirements - prerequisites

Before running WaSiM on any system, there are some requirements to be regarded:

for MPI-versions: An MPI-framework like MPICH2 on Windows or mpi-1.1 packages on Linux must be installed. The MPI-versions of WaSiM are still experimental but can be provided on request (not all advantages of MPI can be used yet and not all sub models and features of WaSiM will be parallelized for MPI).

For all Windows Versions using OpenMP the Microsoft Visual C++ 2013 runtime must be installed (start search in your favorite search engine by typing in
"Microsoft Visual C++ 2013 Redistributable Package (x64)" for Windows 64 bit and
"Microsoft Visual C++ 2013 Redistributable Package (x86)" for Windows 32 bit or go to Microsoft's download page and type in the respective search topics). Parallel installation should be possible as well. For versions before 9.08.03, the redistributables for Visual C++ 2010 must be installed instead.

On Linux computers, there should the GNU compiler suite version 4.2 or higher be installed. Otherwise, the OpenMP versions will not work.

4.2 WaSiM runs from the command line with a control file

No matter on which computer platform WaSiM is started - it is always a command line application which must be run by either using a batch file or the command line (there is one exception: a GUI version of WaSiM exists for the Richards-version, but this GUI version is still not fully functional and can be seen as a demo what WaSiM would look like as a GUI application: the graphical user interface allows the plotting of time series during the model run in various charts).

To start WaSiM on a Windows machine, a command prompt should be started first (Start → cmd.exe or start → run → cmd.exe). Under Linux, a terminal console should be started. However, although it is possible to start WaSiM by double clicking on a control file if this file type was registered accordingly, it is not recommended to do so since usually the command window will close after the execution finishes. In (the really rare ;-) case of errors this is inconvenient since the user must open the XML-log file first to see what went wrong.

The general command syntax is:

```
<programname> <controlfile1> [<controlfile2> [...]] [-diaglevel:FATAL|ERROR|WARNING|INFO|
DEBUG] [-diagfile:<diagnostics file>]
```

There can by any number of control files given as parameters. WaSiM will run them one after the other. This can also be achieved by calling WaSiM multiple times from a batch file. There is no functional difference except that the diagnostics file is re-created each time a new wasim-instance is started, whereas the same log file will be used for all control files when WaSiM is started with multiple control file parameters.

The executable names are defined as follows:

Windows:

wasim1.exe	Richards, graphical version for Windows using OpenMP (only 32 bit)
wasimvc32.exe	Topmodel, Command line version for Windows 32 bit
wasimvco32.exe	Topmodel, Command line version for Windows 32 bit using OpenMP
wasimvcz32.exe	Richards, Command line version for Windows 32 bit
wasimvzo32.exe	Richards, Command line version for Windows 32 bit using OpenMP
wasimvc64.exe	Topmodel, Command line version for Windows 64 bit
wasimvco64.exe	Topmodel, Command line version for Windows 64 bit using OpenMP
wasimvcz64.exe	Richards, Command line version for Windows 64 bit
wasimvzo64.exe	Richards, Command line version for Windows 64 bit using OpenMP
wasimmpi32.exe	Richards, Command line version for Windows 32 bit using MPI
wasimmpi64.exe	Richards, Command line version for Windows 64 bit using MPI

Linux:

wasimtop-<version>-x86-64-<linux-version>	topmodel (compiled on the specified <linux-version>) on Intel/AMD-PC with 64-bit architecture, using OpenMP
wasimtop-<version>-i586-<linux-version>	topmodel (for the specified <linux-version>) on Intel/AMD-PC with 32-bit architecture, using OpenMP
wasimuzr-<version>-x86-64-<linux-version>	Richards (for the specified <linux-version>) on Intel/AMD-PC with 64-bit architecture, using OpenMP
wasimuzr-<version>-i586-<linux-version>	Richards (for the specified <linux-version>) on Intel/AMD-PC with 32-bit architecture, using OpenMP
wasimmpi-<version>-x86-64-<linux-version>	Richards for Linux on Intel/AMD-PC with 64-bit architecture, using MPI

examples:

wasimtop-9-00-07-x86-64-pc-linux-ubuntu → topmodel-version, 64 bit on Ubuntu

wasimuzr-9-00-07-x86-64-pc-linux-ubuntu → Richards-version, 64 bit on Ubuntu

When WaSiM runs, all modules will be initialized first. If an error occurs, the model stops with an explaining error message. Since reading from and writing to hard disks can be very time consuming, WaSiM uses internal buffers for both reading and writing files. Therefor it can take a rather long time until results are written to the result statistics. However, the model run can be manually interrupted by pressing Ctrl+C. WaSiM will finish the actual time step and present a choice how to react: by pressing either

- 0 (nothing),
- 1 (finish model run after next time step),
- 2 (stop immediately, no results will be written!) or
- 3 (flush buffers to disk only and go on)

the user can interactively decide what to do. When pressing 3, the statistic output files will be written (only to the actual time step, of course) and the user can then already load these files into any software visualizing time series (e.g. Graphlines or GNU-plot etc.)

4.3 General structure of the control file

This section describes the general structure of the control file. However, the specific formats of the sections (like e.g. the soiltable, the landuse_table, routing descriptions etc. are described in the respective sections. A complete control file can be found in the appendix.

4.3.1 Structure of the control file

The control file contains all information required for the model run. There are three types of information:

- parameters controlling the model run (grid write codes, statistic file write codes, output lists etc.)
- file names and path names for the input and output streams
- hydrologic model parameters, property tables (for land use, soil types, tracers), static model parameters

The three parameter types are usually mixed in the control file, because it is organized in a thematic way. The control file is subdivided into separate sections, each section responsible for a submodel or for a separate theme, like for an interpolation parameterization or for the land use table. The syntax of the control file is shown in the following example:

```
# comment (optional)

[first_section]
<first Entry>    # comment (optional)
<second Entry>   # comment  --- " ---
<third Entry>    # comment  --- " ---

# comment (optional)
# comment  --- " ---

[second_section]
<first  Entry>   # comment (optional)
<second Entry>   # comment  --- " ---
# comment (comments are removed in a first interpretation step)
<third Entry>    # comment  (optional)
<fourth Entry>   # comment   --- " ---

[third_section]
<first Entry>    # comment   --- " ---
<second Entry>   # comment   --- " ---
.
.
```

Some parameters, like the parameters for soils and land uses, are arranged in tables. Those tables have their own syntax using { and } and closing each parameter record with a semi-colon etc. The respective sections of this documentation will explain the specific syntax.

the syntax of the control file:

Sections starts with a section header consisting of key words and embraced by brackets “[…]”. The model is able to identify a section by the section header only. Although the sections may appear at any position in the control file it is recommended to organize the order of the sections following e.g. the order of the submodels. Within the sections the order of the entries is fix. Between entries no empty rows are allowed but there may be comment rows, which must start with a “#”-character. Each row consists of one entry which may consists itself of one or more parameters, file names or combinations of both. Following the parameters and/or file names of an entry, a comment may follow which doesn’t necessarily have to start with a “#”. The comment has to be separated by the last parameter by a space or by a tab stop.

4.3.2 Using Comments

Between sections, comments may appear in any number. Such comments should also be started using the "#"-character. Only within sections comments must be prefixed with a "#". Comments which start with a "#" are removed from the control file during the model initialization, so also comment rows within sections are removed.

Comments can also be used in parameter lines after the last parameter. The "#" is not required, but is is strongly recommended to mark comments as such in order to avoid strange errors due to unremoved comments.

4.3.3 Using variables with the control file

Because some parameters or parts of file names may appear very often in the control file it is possible to declare up to 2000 symbols. These symbols will be replaced by their content during the translation of the control file. The syntax of a symbol declaration is:

$set $symbol = symbol_value

The declaration starts with a "$set" at the start of a row (without preceding spaces or tabs). The symbol itself starts also with a Dollar-sign "$". The symbol name is followed by an equal sign which is followed by the symbol value which is always interpreted as a character string. Each of the four elements of such a declaration must be separated from the others by at least one space or tab stop. This symbol may then be used instead of the symbol value itself at any location in the control file after the declaration. The symbol value may contain also symbols, which are replaced by their value if the symbol which contains the other symbol is declared.

A symbol in the control file must be separated from the preceding and trailing text by spaces or tabs. If it is required that the symbol values of two or more symbols are interpreted as parts of a longer entry, e.g. a path, a file name and a file extension, this would result in a string including some spaces which should not be there. Therefore it is possible to link the symbol values using the concatenation operator "//" which is used instead of a separator.

Example for the replacement of symbols in the control file:

```
# 1st declaration for the path of the input data
$set $path = c:\wasim\input\

# 2nd declaration
$set $grid = m500

# 3rd declaration
$set $elevationmodel = $grid//.dhm
.
.
.
# somewhere in the control file there is a section containing only
# the name of the elevation model

[elevation_model]
$path//$elevationmodel # name of the digital elevation model
```

The value of the symbol $elevationmodel is `m500.dhm`, because after declaring $elevationmodel the symbol "$grid" is replaced by its value "m500" and linked to the extension ".dhm" because of the "//"-operator. If a "$"-character is found in the control file without a "$set" command at the start of the row, the usage of a symbol is expected (this implies, that no other "$"-characters are allowed in the control file except for declaring or applying symbols). If the symbol is not listed in the symbol table which is generated in a first scan of the file, the model stops with an error message.

Parsing the section [elevation_model] in the above example the program finds the text

`$path//$elevationmodel`. The operator "//" marks the end of a first symbol "$path". This symbol is replaced by its value. Afterward the row looks like this (comment was removed as well):

```
c:\wasim\input\//$Hoehenmodell
```

A second scan of the row gives the result that there is another symbol `$Hoehenmodell` used. Replacing this symbol by its value and linking the value to the preceding text, the complete row would be:

```
c:\wasim\input\m500.dhm
```

This row does not contain any further symbol. This replacement scheme is applied to all rows of the control file before starting the model initialization. The translated control file is stored in a file named like the control file but extended by "$$$". This file contains no symbols and no comments (note: only comments starting with a "#" were removed). The translated control file is then used for model initialization.

4.3.4 Mandatory sections

The following sections must be present in every control file (if not, an error message will appear and the model will stop right on start up. If a submodel is not used, it is usually sufficient to set the first parameter to 0 (the activation-switch for the submodel). See the control file reference for detailed explanations.

common sections:

```
[output_list]
[output_interval]
[coordinates]
[region_transition_distance]
[soil_surface_groundwater_substeps]
[elevation_model]
[zone_grid]
[standard_grids]
[variable_grids]
[model_time]
[meteo_data_count]
[meteo_names]
```

names of the section headers for interpolations like defined in the section [meteo_names] (examples only):

```
[temperature]
[wind_speed]
[precipitation]
[ sunshine_duration]
[global_radiation]
[air_humidity]
[vapor_pressure]
[tracer_input_1]            (only if tracers are used)
.
.
[tracer_input_9]            (only if tracers are used)
```

now the names of the submodels:

```
[RegionalSuperposition]
```

```
[precipitation_correction]
[radiation_correction]
[evapotranspiration]
[snow_model]
[ice_firn] (optional)
[permafrost] (optional, deprecated)
[interception_model]
[infiltration_model]        (required only in WaSiM-Version 1)
[soil_model]                (required only in WaSiM-Version 1)
[SiltingUpModel]            (required only in WaSiM-Version 2)
[SurfaceRoutingModel]       (required only in WaSiM-Version 2)
[unsatzon_model]            (required only in WaSiM-Version 2)
[irrigation]                (required only in WaSiM-Version 2)
[groundwater_flow]          (required only in WaSiM-Version 2)
[lake_model]
[ExternalCoupling]
[routing_model]
```

If the model uses abstractions (from rivers), each abstraction does need a following section:

```
[abstraction_rule_abstraction_1]
...
[abstraction_rule_abstraction_x]
```

If the routing module uses reservoirs, for each reservoir a section with an abstraction rule must be present (x stands for the number of the reservoir and should not appear as “x” in the real control file):

```
[abstraction_rule_reservoir_1]
.
.
[abstraction_rule_reservoir_x]
```

and finally the sections containing land use dependent and soil type dependent properties:

```
[multilayer_landuse]
[landuse_table]
[soil_table]
[substance_transport] (required only in WaSiM-Version 2)
[irrigation_table]     (required only in WaSiM-Version 2)
```

4.4 Using the XML log file

WaSiM automatically writes an XML diagnostic file. The default settings (if no command line argument for the file name and the log-level were given) are:

- filen ame: WaSiM_Diag.xml
- diagnostic level: DEBUG

By using the optional command line parameters

```
[-diaglevel:FATAL|ERROR|WARNING|INFO|DEBUG] [-diagfile:<diagnostics file>]
```

it is possible to define other filenames and levels. Possible levels and their representation in the XML as attributes “level” are:

- DEBUG (level=”4”)
- INFO (level=”3”)
- WARNING (level=”2”)

- ERROR (level=”1”)
- FATAL (level=”0”)

All messages with a level less or equal to the defined diaglevel will appear in the XML file.

A minimal XML-diagnostic file looks like this:

```
<?xml version='1.0' encoding='UTF-8'?>
<Diag version="1.2" xmlns="http://www.wldelft.nl/fews/PI"
xmlns:xsi="http://www.w3.org/2001/XMLscheme-instance"
xsi:schemeLocation="http://www.wldelft.nl/fews/PI
http://fews.wldelft.nl/schemes/version1.0/pi-schemes/pi_diag.xsd">
       <line level="3" description= "WaSiM start at 2012-03-31T18:54:42"/>
       <line level="3" description= "WaSiM finished at 2012-03-31T18:54:59"/>
</Diag>
```

As can be seen in the example, the first and last entry will be always written, even if the diaglevel-parameter was set to values less than 3. When using the default (diaglevel = DEBUG (level=”4”)), the XML file contains the output which is also printed to the console.

4.5 Recommended Directory Structure

Though it is not necessary to have a directory structure at all it is yet highly recommended to structure the data for a project. The minimum structure should consist of the following directories:

- a main directory for the project or the project variant
  - an input directory
  - an output directory (one separate directory per model run)
  - an optional initialization directory
  - an optional exchange directory

This directories may of course be further refined by the user.

4.5.1 Input directory

Here, all meteorologic inputs, hydrologic observations (observed discharges) and all static input grids should be placed. Optionally, each of the mentioned data categories may be placed in a separate sub-directory.

4.5.2 Default Output directory

The control file allows the definition of a parameter called “DefaultOutputDirectory”. It may contain a complete path (including drive letter or root or home directory). If this parameter is defined in the section [output_interval], then all output file names which do not have a drive letter (like c: or d: etc. under Windows) or a leading / or ~ under Linux at the starting position will be extended internally by the value of this parameter. Any file name containing a drive letter or a root or home directory is not changed at all. The default output directory may not only contain an absolute path but also a relative path.

Example for the definition of paths:

```
$set $basepath                =  d:\Data\m500Opt\
$set $outpath                 =
$set $inpath                  =  $basepath//input\
$set $exchngpath              =  $basepath//exchng\
$set $InitialStateDirectory   =  $basepath//StateIni\
$set $DefaultOutputDirectory  =  $basepath//output\

[output_interval]
```

```
50          # increment of time steps until an output to the screen is done
1           # warning level for interpolation (no station within search radius)
0     # output of runoff results in 0=mm/time step, 1=m3/s
0     # minutes from the hour-entry in the input data files to the end of the
# timestep
WriteAsciiGrids = 0
InitialStateDirectory = $InitialStateDirectory
DefaultOutputDirectory = $DefaultOutputDirectory
```

The empty $outpath variable is for compatibility with old control files. It makes sure that the file names have no prepending absolute path, so the value of DefaultOutputDirectory will always be used for writing files, in this case the output path will be d:\Data\m500Otp\output.

4.5.3 Initial State Directory

The parameter InitialStateDirectory works similar to DefaultOutputDirectory. If this parameter is found, the value will be prepended to all grids and other files that will be used for initialization, but only if those file names are provided without leading absolute paths. So the value of InitialStateDirectory is used to generate the complete file name for initialization and the value of DefaultOutputDirectory is used to generate the complete file name for writing. Thus, the same file (e.g. groundwater table depth) can be read from another place than it is written to. This is not possible when using none of the parameters InitialStateDirectory and DefaultOutputDirectory, since then both names (for reading and writing) would be identical.

In other words: by using InitialStateDirectory and DefaultOutputDirectory it is possible to use constant initialization grids for a series of model runs while all the result grids may be different. If a certain set of grids from a distinct model run should be used for initialization of another model run, the respective grids can simply be copied from the output-directory of the old model run to the initial state directory given in InitialStateDirectory.

The files contained in the initialState directory are

- all state grids, like
  - actual soil water content (theta)
  - actual hydraulic heads (hhyd)
  - actual geodetic heads (hgeo)
  - snow storages (solid, liquid, on and off glaciers, ttoal SWE)
  - temperature stacks
  - groundwater level
  - groundwater depth
  - root depth
  - interception storage
  - …
- storage_richards.ftz_0 … storage_richards.ftz_n; these files contain all non-grid storages that are required to initialize a subsequent model run with values of the preceding run so that there is a smooth join:
  - storages from all routing channels in section [routing_model] or [routing_model_v2]
  - storages for runoff concentration single linear reservoir cascades for direct runoff and interflow in section [unsatzon_model] or [unsatzon_model_v2]
  - single linear reservoir content for glacier runoff components per subbasin in section

[snow_model]

4.5.3.1 Seamless coupling of model runs with different time steps

Using sections [unsatzon_model_v2] and [routing_model_v2], WaSiM is able to convert the initialization values for runoff routing (channel routing in module routing_model) and runoff concentration (surface runoff and interflow concentration within a subbasin in module unsatzon_model) from files that were saved for another internal data structure. Still, the routing structure in section [routing_model] must be the same (i.e. number of routings, the structure of which tributary flows into which basin, abstractions, inflows etc.), but the hydraulic parameters can vary: length, roughness, geometry (width and depth of main channel and flood plains), storage coefficients and slopes can be changed. Also the basic parameters for computing the waterlevel-runoff-tables (i.e. minimal and maximal specific discharge, number of logarithmic steps and numbe rof sub time steps) can be different than in the preceding run. If the storage_richards file doesn't contain the new but only the old section names, then the reading mechanism uses the old methiod to make the model backwards compatible. That means, that old states can be processed by newer model versions, but newer model versions output can't be processed by older model versions. If this should be required, the solution is simple: remove the line with the comment "(timestep in minutes)" from the file and change the sections [unsatzon_model_v2] and [routing_model_v2] to [unsatzon_model] and [routing_model].

[routing_model_v2]

All the above mentioned parameters can change the internal routing structure either globally (basic parameters) or per channel. Internally, each channel is represented by a series of subchannels, which are subdivided in more or less sub time step parts, the number depending on how long the water needs to flow through the entire channel. So the internal structure depends on the hydraulic parameters. If a model starts with changed parameters compared to the initialization file, it is now able to recognize the original structure (by a few additional parameters in the _v2-versions of the sections). It will then not read in the complete structure as expected, but rather converts the read in values to the actual structure created from the actual parameters. To keep it simple, the conversion doesn't match any specific sub time step and sub channel routing element to the actual structure, but matches the first elements only using simpel integer division.

Example: if an original channel was subdivided into 5 sub channels with 30 sub elements each and the original sub time step was 2 minutes for an interval (time step) of 1hour, whereas the new parameters for a 10 min time step lead to 34 sub channels with 15 sub elements each and a sub time step of 2.5 minutes, a matching must be found to map the 34*15 = 510 new routing elements to the 5*30 = 150 routing elements from the input. Since the routing uses an internal array offset for the kinematic wave approach, it is not save to assume that all existing routing sub elements were actively used for routing. The implemented algorithm rather chooses the simple but effective approach to take the first value from the first input sub channel and assign it to all sub elements of the first 34/5 = 7 (rounded integer division) sub channels in the actually created routing structure. Here, the matching would be like this:

1st element of 1st source sub channel –> all elements of 1st to 7th target sub channels (7)

1st element of 2nd source sub channel –> all elements of 8th to 14th target sub channels (7)

1st element of 3rd source sub channel –> all elements of 15th to 20th target sub channels (6!)

1st element of 4th source sub channel –> all elements of 21st to 27th target sub channels (7)

1st element of 5th source sub channel –> all elements of 27th to 34th target sub channels (7)

Thus, variations within the sources sub channels will not be mapped to the target elements, but all tests showed that even in situations when one model run starts within a steeply increasing flood

wave, that the differences between a continuous run and a coupled run are almost invisible.

[unsatzon_model_v2]

The main ans single difference between old and new version is the additional parameter "timestep in minutes" as second parameter in section [unsatzon_model_v2]. It is used to separate or accumulate read values to the actual single linear reservoir chain structure for surface runoff and interflow concentration within each subbasin. All other parameters of [unsatzon_model] must still be unchanged in the actual model run with respect to the source storage_richards file (i.e. number and codes of sub basins, storage constants for interflow and direct runoff, drainage density, baseflow and snow melt runoff fraction parameters).

This functionality was implemented to be able to couple runs with different time steps. If the first run had a time step of 1 hour, all travel time zones for interflow and direct runoff were parameterized to match one hour flow time per time zone. For a subsequent run woth e.g. a 10 min time step, there will be around 6 times as many time zones with 10 min travel time each.

The mapping between source and target is done using an area weighed accumulation or diverting algorithm, making sure that the total amount of stored water is identical to the source configuration. Each zone of the actual structure gets an amount of the matching source zone that is equivalent to it's relative area on the original zone.

The last Step of converting the storage contents is to scale the actual outflow of the direct runoff and interflow storage by the relation between the old and the new time step (i.e. 1 h –> 10 min: apply a correction factor of $1/6^{th}$).

Example: Figure 4.5.3.1 shows the effect of this coupling: the red graph, that is hidden behind the blue and green graph, is the continuous simulation. Blue is the simulations that stops at May 30 th , 24:00. Grenn is the continuing model run that uses the same time step and hence could read in the storage file content in old style. The black graph, however, is the result of a 10-min-time step model run. As can be seen, the starting point is exactly where it should be and the shape of the flood wave is also quite similar to the original run.

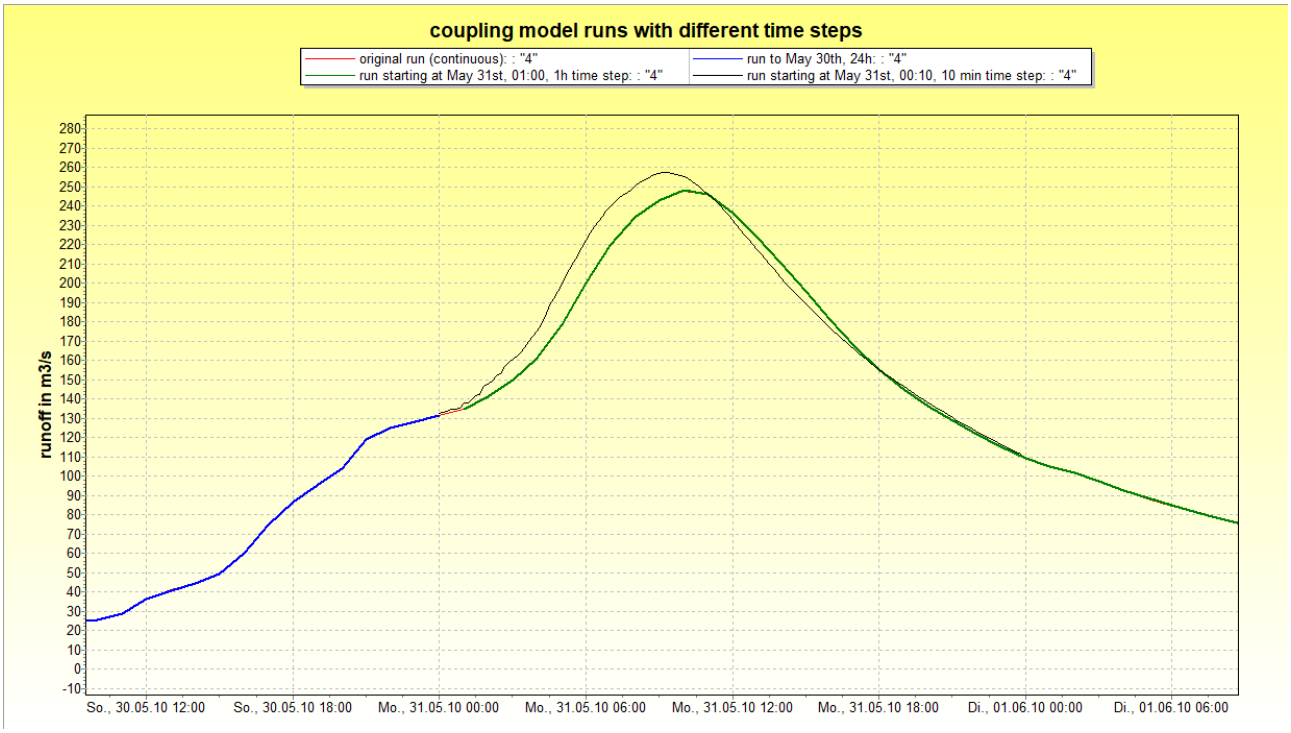


figure 4.5.3.1: Effect of new initialization of runoff concentration and runoff routing when coupling runs with different time steps (here: green uses 1h, black uses 10min time step)

The somewhat steeper increase of the flood wave and the earlier occurrence of the maximum is mainly an effect of different routing parameters, which must slightly be recalibrated usually. However, it is also an effect of another time step offset: 10 min models always use 00:00 as base time (i.e. time range is from 00:00 to 23:59, or 00:00 to 23:50 with 10 min time steps), whereas with 1 hour time step it is possible – and widely used – to use the hour numbers from 1 to 24. The green time series (coupled run with 1 hour time step) uses this offset of 1 hour, whereas the 10 min run uses zero offset. This should lead to a 10 min offset between both coupled runs.

4.5.4 Exchange Directory

An exchange directory is required when using WaSiM together with another model, e.g. a groundwater model. The directory doesn't have to be declared as a variable, but it is good practice to do so. All files referenced in the section [ExternalCoupling] should be prefixed with the variable for the exchange-directory.

4.6 Input file formats

4.6.1 Spatially gridded data

In WaSiM grids can be used in binary or in ASCII-format. In order to exchange grids between different computer platforms (Windows ←→ Linux) or to exchange grids with other software, e.g. with Arc/Info, an ASCII-format is usually required:

rows 1..6: grid header (number of columns and rows, coordinates of the lower left corner, cell size, nodata value)

the following rows will be the grid data from top left to down right like in a book:

```
ncols            7
nrows           12
xllcorner    680500
yllcorner    222200
cellsize       2000
nodata_value  -9999
-9999  -9999  -9999  -9999  -9999  -9999  -9999
-9999  -9999  -9999  -9999  -9999  -9999  -9999
-9999  -9999    586    559    595    601    585
  584  -9999  -9999  -9999  -9999  -9999  -9999
-9999    449  -9999  -9999    454    511    551
-9999    561    529    509    498    487    499
  521    523    523    534  -9999  -9999  -9999
-9999  -9999  -9999  -9999  -9999  -9999  -9999
-9999  -9999  -9999  -9999  -9999    345    234
  567    453    456    478    789    576    482
  123  -9999   1252   1211    876    564    825
-9999  -9999    234   1246  -9999  -9999  -9999
```

The binary format is structured similar to the ASCII-format. All values are stored in single precision (4 bytes in most computer systems, 6..7 digits precision). Records 1 to 6 are the same data like in the ASCII-Format. Then 6 additional records describing the grid statistics follow:

Records 1..6: like in the ASCII-format
Records 7..12:

- count of valid cells (without nodata-values)
- minimum value of all valid grid cells
- maximum value of all valid grid cells
- sum of all valid grid cells
- mean value of all valid grid cells
- standard derivation of all valid grid cells

After this header all cells of the grid are following in the same order like in the ASCII-format, i.e. from the top left cell to the lower right cell in a row by row order. For converting the formats one into the other there are the programs ASCIGRID and GRIDASCI. The described formats are the same for all system platforms. However, there are internal differences in storing binary numbers dependent on the processor layout. Also ASCII-data are stored in different ways on different systems, but ASCII-formats can be exchanged between different systems using e.g. ftp-programs with an automatic conversion of carriage returns/line feeds. To exchange binary data, they first have to be converted into ASCII-data, then the exchange can be done and thereafter a re-converting on the target platform should be done.

GIF-Input:

As a special input format, GIF files can be read in by meteorologic interpolators. Each grid must be contained in a single gif file, the list of files must be defined in an input text file like the ones used for meteorologic station data (see next chapter). More information on using GIF input can be found in Chapter 2.3.9 .

netCDF-Input:

Also as a special input format, the netCDF format can be read in by the interpolation methods. Standard grids could also be read in if the necessary extensions to the filename are found by the reader. *Starting with version 10.06.00, standard and variable grids as well as stacks can be read in without extensions to their names, so it is easier to use netCDF output grids as initialization in ther StateIni folder of the coupled run. However, if netCDF inputs are used for meteorologic interpolation with many grids as time series within a single netCDF, the following extension MUST be used in order to avoid reading the first layer in the netCDF each time a new time step should be read in*:

filename<variable>layer

Example:

hurs_observations_historical_19700101-19701231.nc<hurs>0

Here, hurs is the variable name that must match the variable name in the netCDF file to be read. 0 is the layer in the netCDF file, used for netCDF files that contain several layers for the same variable. The typical structure of an input netCDF may look like this (in ASCII representation, created by ncdump):

```
netcdf  pr_bav-EUR-11_observations_historical_NA_LfU-KliRef2014_v1_day_19700101-
19701231 {
dimensions:
      y = 37 ;
      x = 34 ;
      time = 365 ;
variables:
      double lon(x, y) ;
            lon:units = "degrees_east" ;
      double lat(x, y) ;
            lat:units = "degrees_north" ;
      double time(time) ;
            time:calendar = "standard" ;
            time:units = "days since 1970-01-01 00:00:00" ;
      double pr(time, x, y) ;
            pr:standard_name = "mm" ;
            pr:long_name = "precipitation" ;
            pr:units = "mm/d" ;
            pr:scale_factor = 1. ;

// global attributes:
            :Na = "NA" ;
```

```
            :history = "Spatial Selection (with MATLAB) based on a vector shape
file; Spatial Selection (with MATLAB) based on a vector shape file" ;
            :Conventions = "na" ;
data:

 lon =
  7.82679891586304, 7.85027408599854, 7.87364196777344, 7.89690446853638,
    7.92006206512451, 7.94311571121216, 7.96606588363647, 7.98891353607178,
    8.01165962219238, 8.03430557250977, 8.05685138702393, 8.07929801940918,
    8.10164737701416, 8.12389850616455, 8.14605236053467, 8.16811180114746,
    8.1900749206543, 8.21194458007813, 8.23371982574463, 8.25540351867676,
    8.27699375152588, 8.29849338531494, 8.31990242004395, 8.34122180938721,
    8.36245250701904, 8.38359355926514, 8.40464782714844, 8.42561435699463,
    8.44649505615234, 8.46728992462158, 8.48799991607666, 8.50862503051758,
    8.52916717529297, 8.54962539672852, 8.57000255584717, 8.59029674530029,
    8.61051082611084,
… (lon for all columns)

 lat =
  50.2489356994629, 50.1399688720703, 50.0309944152832, 49.9220161437988,
    49.8130340576172, 49.704044342041, 49.5950546264648, 49.4860572814941,
    49.3770561218262, 49.2680511474609, 49.1590423583984, 49.0500297546387,
    48.9410095214844, 48.8319892883301, 48.7229614257813, 48.6139297485352,
    48.5048942565918, 48.3958549499512, 48.2868118286133, 48.1777648925781,
    48.0687103271484, 47.9596557617188, 47.8505973815918, 47.7415313720703,
    47.6324653625488, 47.5233955383301, 47.4143180847168, 47.3052406311035,
    47.1961555480957, 47.0870704650879, 46.9779777526855, 46.8688850402832,
    46.7597846984863, 46.6506843566895, 46.5415802001953, 46.4324722290039,
    46.323356628418,
… (lat for all columns)

 time = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
    20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,
… (all time steps)

 pr =
  1.1858035326004, 0.629651010036469, 0.286435157060623, 0.163546726107597,
    0.349319189786911, 0.307044684886932, 0.36788022518158,
    0.318119317293167, 0.83652925491333, 0.225907787680626,
    0.383342057466507, 0.863127946853638, 0.73056560754776,
    0.226972550153732, 0.185827076435089, 1.34628689289093,
    0.552619695663452, 0.57875382900238, 0.283306419849396,
    0.442573308944702, 0.29509037733078, 0.572198569774628, 1.38391542434692,
    0.00916849263012409, 0.0189180113375187, 0.155018821358681,
    0.19671918451786, 0.636679768562317, 0.376025944948196,
    0.836916744709015, 0.810808002948761, 0.992812216281891,
    1.28943169116974, 1.09801197052002, 1.70412135124207, 1.25278294086456,
    0.482728004455566,
… (all columns and then all time steps for variable pr)
```

4.6.2 Meteorologic time series

Meteorological data as station data are stored in text files. The first 5 rows contain descriptive data like station coordinates, altitudes, and station names. The station files are organized by columns, i.e. each station is represented by a column in the file. The first 4 data columns contain the date and time of the data. The columns appear in the order year – month – day – hour. The hour may either be the number of the actual hour, then e.g. „1984 1 2 5“ is the time step from 4:00 to 5:00 at the 2nd of January in 1984, or the hour may be the start of the hour, then „1984 1 2 5“ would be the time step from 5:00 to 6:00 at the 2nd of January in 1984. For daily time steps the hour can also be set to either 0 or 24. All station files must share the same hour range (e.g. either 0..23 or 1..24). The type of the hour range is coded in the control file as fourth parameter in the [output_interval]-section.

The parameter must be zero, if the range is 1...24 (or 24 for daily time steps and longer) and it must be the duration of a time step in minutes, if the hour range is 0...23 (or 0 for daily time steps and longer), e.g. 60 for hourly time steps and 1440 for daily time steps. As delimiters between the columns tab stops are recommended. Spaces are allowed as well. Multiple spaces and tabs are interpreted as a single white space. The year can be written in the format YYYY or in the format YY, which is assumed to be valid for the 20th century only, so YY is internally extended to 19YY.

description of meteorological/hydrological input data:

Row 1: comment
Row 2: after „yy mm dd hh“: altitudes for each station (int or float), basin area for hydrologic data
Row 3: after „yy mm dd hh“: x-coordinates of the stations (integer or floating point values)
Row 4: after „yy mm dd hh“: y-coordinates of the stations (integer or floating point values)
Row 5: after „yy mm dd hh“: short identifier for each station e.g. 6-chars
beginning with Row 6: actual date (e.g. 1984 01 01 24), then for each station one value (real or integer) separated by at least one space or tab stop.

Example of a file containing meteorological station data (temperatures of ANETZ-station for 1984):

```
temperatures_(*0.1C)
YY MM DD HH     437    440    463    515    536    750    779   1407   1590   2490   2690
YY MM DD HH  688700 738430 693770 723750 710500 718500 747940 638130 783580 744100 780600
YY MM DD HH  282800 273950 230780 210580 259820 248000 254600 205970 187480 234900 189630
YY MM DD HH  Schaff Guetti Waeden Glarus Taenik Rietho St.Gal   Napf  Davos Saenti Weissf
84  1  1  1      17      6     25     28     37     49     39     33    -13    -31    -34
84  1  1  2      13      1     26     25     42     51     45     31    -10    -29    -27
84  1  1  3       8     -5     22     23     49     60     47     32     -5    -25    -25
84  1  1  4       8     -5     14     27     41     36     47     42     -2    -23    -18
84  1  1  5       6     -8     10     22     21     13     38     53      1    -17    -20
```

Meteorological input data may also be interpolated by altitude dependent regression. This is done during the preprocessing using the programs “regress” (a graphical DOS-program) or “regr” (a console version of “regress”). The program “regress” generates ASCII-output which is also directly useable by WaSiM. The model recognizes the format by analyzing the header. If the first characters are “jr mo tg st” (separated by single spaces!), then an ASCII-format is assumed, otherwise a binary format is assumed. Both formats have the same structure except that the binary format does not have a header. The binary file contains all data as single precision values (4 byte per value), the first four data columns for date and time are stored as long integers (4 byte). An example of such a regression file is given below:

```
jr mo tg st     a[1]     b[1]  igu    a[2]   b[2]   igo     a[3]     b[3] af        bf[1]        bf[2]
84  1  1  1  -17.954  0.0000   0.0 -17.954 0.0833  969.2 117.001 -0.05591 51.126   -7.21913E-05 -7.45598E-05
84  1  1  2  -29.724  0.0000   0.0 -29.724 0.1048  872.6 105.243 -0.04981 39.8687  -6.95074E-05 -8.48521E-05
84  1  1  3  -46.493  0.0000   0.0 -46.493 0.1350  822.3 104.321 -0.04835 30.1048  -8.14217E-05 -7.85187E-05
84  1  1  4  -34.818  0.0000   0.0 -34.818 0.1050  970.4 117.607 -0.05205 13.8404  -3.43135E-05 -0.000102529
84  1  1  5  -23.550  0.0000   0.0 -23.550 0.0680 1275.7 140.102 -0.06021 11.6198  -1.80593E-05 -9.44962E-05
84  1  1  6  -44.244  0.0000   0.0 -44.244 0.1083 1103.7 140.419 -0.05898 -26.0965     -6.67409E-07 -7.49243E-05
84  1  1  7  -25.346  0.0602 820.0 -17.291 0.0504 1400.0 125.643 -0.05163 -7.7026  -1.50661E-05 -2.84531E-05
84  1  1  8  -17.163  0.0387 820.0 -40.929 0.0677 1400.0 122.876 -0.04930 -17.3485  2.76033E-06 -7.57734E-06
84  1  1  9  -28.146  0.0677 820.0  -7.093 0.0420 1400.0 114.150 -0.04457 -7.1713  -2.59317E-05 -9.54698E-06
84  1  1 10  -14.794  0.0514 820.0  -6.893 0.0418 1400.0 106.073 -0.03888 14.4881  -3.8284E-05  -7.03115E-06
84  1  1 11  -22.906  0.0934 820.0  32.876 0.0253 1400.0 138.656 -0.05017 -15.7421     -1.69895E-05  2.13564E-05
84  1  1 12  -18.494  0.1048 820.0  49.093 0.0224 1400.0 163.018 -0.05894 -5.4302      -2.60689E-05  2.40711E-05
```

description:

jr, mo, tg, st: this rows contain date and hour, for which the following gradients are valid (each a 4 byte long integer). separated each by **only one space**, hours must be given like in station data files as 0...23 or 1...24 consistent to the station files which are used at the same time

a[1], a[2], a[3]: constants of the linear equation $y = a + bx$ for the lower, the middle and the upper line.

igu:	upper validity limit of the lower regression line
igo:	upper validity limit of the middle regression line and lower validity limit of the upper regression line.
af, bf[1], bf[2]	regression parameter of the areal regression of the residuals in $z = a + bf[1] \cdot x + bf[2] \cdot y$

Note: when using the interpolation methods 10 or 11, no input file with precomputed gradients is required, since the gradients will be estimated directly in WaSiM during the model run.

Note 2: When using a smaller time step than one hour, then the hour entry may be simply used again. Running WaSiM in 10 minute time steps would mean to use any year-month-day-hour header 6 times for each hour. WaSiM will recognize the time step by a parameter provided with each sub model, so the meteorologic input data only needs to refer to the integer hour.

4.6.3 Hydrological time series

Hydrological data are time series of observed discharges at river gauges. These data are required in order to calibrate the model and to estimate the model efficiency. The format is identical with the format for meteorological station data, except for the second row which does not contains the altitude but the catchment area of a gauging station. This is important if the discharges (in m^3/s) are converted into specific discharges (in mm/time step) using the program QTOSPEND. The unit of the discharge file must be mm/time step.

4.7 Reading old states as initial conditions

Initial conditions may be very important for the model results, especially if applying the model to a short time period or if the internal storage's differ considerably from their real value (e.g. groundwater table much higher or much deeper than in reality). Therefore the possibility to read storage or reservoir content values as initialization grids is implemented in WaSiM. Because these grids can also be written at the end of a model run, they can be used as initialization for following model runs. If no initialization grids are available, the internal states are initialized using constant values taken from the control file. Initialization grids and result grids (actual states and reservoir contents) either must have the same names and they must be located in the same directory – or DefaultOutputDirectory and InitialStateDirectory parameters are used. When not using the latter possibility, the initialization grids will be overwritten by the result grids.

4.7.1 Grids to be read

Grids which can be used for initialization are listed in the following overview (in brackets: file names as used in the sample control file):

grids for multiple submodels

- soil moisture storage (SB) [1/1] (only for model version 1)
- albedo (ALBE) [1/1]

soil model (version 1 with TOPMODEL-approach)

- saturation deficit (SD) [mm]
- storage content of the unsaturated soil zone (SUZ) [mm]
- interflow storage (SIF) [mm]
- soil moisture in the inactive soil storage (SBia) [1/1]
- available field capacity in the inactive soil storage (nFKia) [mm]

- areas which are actually saturated by infiltration after GREEN and AMPT (SATT) [-]
- fraction of the soil moisture storage which is in hydraulic contact to the groundwater (SSP) [-]

soil model and groundwater model (version 2 with RICHARDS-approach)

- water content off all discretization layers (THET → Stack) [1/1]
- hydraulic heads (potential) of all discretization layers (HHYD → Stack) [m a.s.l]
- geodetic elevations of all of all discretization layers (HGEO → Stack) [m a.s.l.]
- water content in ponds (POND) [mm]
- groundwater table (GWST) [m below surface]
- mean water content of the layer, the groundwater is located within (GWTH) [1/1]
- index of the layer, the groundwater is located within (GWLV) [-]
- concentrations of modeled tracers for the unsaturated zone (e.g. NACUNSAT → Stacks) [1/1]
- groundwater heads for all aquifers (GWH1..GWHn) [m a.s.l.]
- concentrations of modeled tracers for the groundwater (e.g. NACGROU1…NACGROU3)

interception model

- content of the interception storage's (per vegetation layer) (SI) [mm]
- concentrations of modeled tracers for the interception storage (e.g. NACINTER, only in version 2)

snow model

- snow fraction in the water equivalent (SSNO) [mm]
- liquid fraction in the water equivalent (SLIQ) [mm]
- total snow storage as water equivalent (SSTO) [mm]
- snow age (SAGE) days without snow [days]
- snow days (SDAY) days with snow [days]
- concentrations of modeled tracers for the snow storage (e.g. NACSNOW_, only in version 2)

heat transfer model

- soil temperatures for all discretization layers (TSOIL → stack) [°C]

If no grids are read in, the storage contents are set to initial values. For snow and interception storage this values are zero. The storage content values of the unsaturated zone and of the saturation deficit can be specified as subbasin specific values in the control file (only version 1). The soil storage is initialized to 1.0 (100 % soil moisture). Using model version 2, the soil moisture is initialized for each discretization layer with a water content corresponding to a hydraulic equilibrium with the groundwater (fluxes are zero). The groundwater table is defined by a parameter in the control file (reciprocal value of that parameter multiplied by the number of numerical layers gives the layer number (counted downward from the soil surface) where the groundwater starts).

For the unsaturated zone it is also possible to read in only the grid containing the groundwater table. Using this grid the initial values for the hydraulic heads in the unsaturated zone are estimated using the hydraulic equilibrium (no flux) condition. This also determines the water content values and the layer index and the mean water content of the layer, the groundwater is located in. All other initialization grids are set to default values (tracer concentrations, pond storage, saturated areas etc.).

Without a soil temperature stack, soil temperatures are initialized by linearly interpolating between the temperature at bottom of the soil (defined as model parameters) and the surface temperature of the first time step (taken from air temperature).

4.7.2 storage content file

When WaSiM executes a series of control files for subsequent time periods which should fit exactly together (i.e., the last discharge value of the first model run and the first discharge value of the second model run are fitting smoothly together), then it´s important to not only save the grids of soil moisture, groundwater table, snow storage etc., but also the actual contents in non-gridded storages like the single linear reservoirs in the different flow travel time zones which are used to route the runoff to the subbasins outlet and the contents of the routing channels. Also, single linear reservoirs of the glacier runoff components must be stored and reloaded accordingly.

Therefore, WaSiM writes a so-called "Storage Content File" containing these data in ASCII-format. The name must be specified as a file name like the other file names without any additional parameters. This name appears after the readgrids-parameter in the sections [soil_model] and [unsatzonmodel] for model versions 1 and 2, respectively.

When starting a new model run, this file will only be read in, if the `$gridread` parameter was set to 1 (or, in case of the [unsatzonmodel], also if it is set to 2, 3 or 4), otherwise the data in this file is ignored, because the grids are not read in, too. If the parameters of the unsaturated zone or soil model in the new model run differ from the ones stored in the "Storage Content File" or if the basin structure has changed (different codes or number of subbasins, other routing parameters), an error will be generated and WaSiM stops it´s execution.

Note: There is a "trick" to make WaSiM to accept changed parameters:

- copy the grids required for initialization (if not sure: copy all grids) to a temporary folder or use the InitialStateDirectory – variable (then, the initial grids will be stored in another folder than the result grids
- start the new model run with readgrids set to 0 and stop it immediately after the first time step. A storage-content-file will be written containing the new parameters.
- Copy this file from the output-directory to the InitialStateDirectory (when using this parameter) or copy the grids from the temporary folder back to the output directory
- change the readgrids parameter to 1
- start the second model run again. Now, the storage grids of the old model run will be read in and the parameters in the storage content file are valid as well – although they may result in sudden changes of model results for the first few time steps (e.g. if changing SLR constants, the runoff may suddenly jump to much higher levels and then slowly return to a lower level).

So if using this trick, the storage grids will be used for initialization but the SLR-values will not be used at all. Depending on the relative importance this may cause strong deviations from the expected runoff, especially for glacier runoff components.

Note: Starting with version 10.06.00, changing time step and routing parameters (not routing structure, only channel properties and w-q-table parameters like min and max specific runoff and number of logarithmic steps to create that table) can be handled by WaSiM automatically. So it is possible to couple runs that use different time steps and/or different hydraulic routing parameters. For more information, see chapter 4.5.3.1 : Seamless coupling of model runs with different time steps .

4.8 Static and dynamic input grids

The control file contains two sections where input grids can be specified. The section [standard_grids] was originally designed to contain a list of grids with constant cell values which will be needed somewhere in the model – e.g. grids describing the basin like digital elevation model, land use data, soil types, aspect, slope etc. Such grids can be called parameter grids. All sub-models can access these grids and their values will not change. Therefor these grids could not be written back to the storage medium when the model finished (this has been changed, as one can probably figure out already, but it's important to understand the original difference between [standard_grids] and [variable_grids].

On the other hand, there are grids which will be used in several sub-models with changing values. For instance the albedo. Such grids could not be defined in a single sub model but must be defined outside (i.e. globally) in order to allow all sub models to access the grid. Since the grid has changing values, it was consequently required to write the grid (with the same possible options like any other sub-model specific grid). The solution was the section [variable_grids]

So there are still both sections for compatibility reasons, but in fact the section [standard_grids] is now offering the same and much more functionality than the section [variable_grids]. It is therefor recommended to use standard_grids whenever possible ([variable_grids] are a kind of deprecated).

4.8.1 Standard grids

A standard grid will be defined in the section standard grids. This section starts with the number of following entries. Each entry has a number of mandatory and optional parameters, explained below. The basic entry with the mandatory parameters looks like this:

```
<filename>     <internal identification>     <fill code>
```

where <filename> is the file name of the grid to be read in, either with full path or with relative path or a filename only: when the parameters DefaultOutputDirectory and InitialStateDirectory are used, WaSiM will create the complete path for such incomplete file names automatically. <internal identification> describes a string which is used internally to identify the grid. For example, the land use grid uses the internal code “landuse” (without quotes). <fill code> is a code defining whether nodata grid cells should be filled using nearest neighbor interpolation for grid cells which are not nodata in the corresponding location of the sub-catchment grid.

There are a number of optional parameters described later.

4.8.1.1 What grids can be standard grids

The following table gives an overview of all grids which may be used as standard grids (others may follow).

Table 4.8.1: list of standard_grids identification codes

Internal identification	Recommen-ded fillcode	comment
landuse	1	Land use codes used as index to the landuse_table; mandatory
zonegrid_soilmodel	1	Zones for the submodels [unsatzon_model], [groundwater_model], [routing_model] and for glaciers in the [snow_model]; mandatory
soil_types	1	Soil types codes used as index to the soil_table; mandatory
flow_times	1	Flow travel times from each cell to the next

Internal identification	Recommen-ded fillcode	comment
		subbasin outlet in seconds; optional (if not provided, no translation is calculated for direct runoff and interflow); optional
river_depth	0	River depth in m for all cells with a river or another natural draining structure (mandatory if interaction with groundwater should be considerd)
river_width	0	River width in m for all cells with a river or another natural draining structure (mandatory if interaction with groundwater should be considerd)
river_links	0	Codes for each routing channel defining the subbasin the channel is coming from (mandatory for computing re-infiltration)
kolmation	0	Colmation of the river links (mandatory for computing re-infiltration)
aquifer_thickness_1*	0	grid with thickness of first aquifer (m from soil surface to the aquifer bottom)
gw_storage_coeff_1*	0	storage coefficients for 1st aquifer
gw_boundary_fix_h_1*	0	boundary conditions constant head for layer 1
gw_boundary_fix_q_1*	0	boundary conditions 2 (given flux perpendicular to the border) for layer 1
gw_k_x_1*	0	lateral hydraulic conductivities for the 1st aquifer in x direction
gw_k_y_1*	0	lateral hydraulic conductivities for the 1st aquifer in y direction
gw_kolmation_1*	0	colmation (leakage factor) between 1st and 2nd aquifer
slope_angle	1	grid with slope angle data in ° (0...90)
slope_aspect	1	grid with slope aspect data in ° (0...360, 0 is North)
RelCellArea	1	Grid with relative cell area as the relation of the actual cell size to the default cell size (when used, also CellSize_X and CellSizeY must be used)
CellSizeX	1	Cell size of each cell in X direction (when used, also CellSizeY and RelCellArea must be provided)
CellSizeY	1	Cell size of each cell in Y direction (when used, also CellSizeX and RelCellArea must be provided)
regression_regions	1	region grid if using multiple regions for meteorological data interpolation and the section [Superposition] is active
ice_firn	0	grid with firn or ice cells, mandatory for static glacier model, **permitted** when using dynamic

Internal identification	Recommen-ded fillcode	comment
		glacier model
FlowDirection	1	grid with flow directions from tanalys (not really used internally)
topographic_faktor	1	soil-topograhic-factor ln(A/(T*tanb)); mandatory for Topmodel-version
hydraulic_conductivity	1	grid with hydraulic conductivity of the soil → Topmodel version only
available_soil_moisture	1	grid with available soil moisture at field capacity [mm] → Topmodel version only
sky_view_factor	1	grid with Sky-View-Factor; used for radiation correction; optional
concflux_tracer_1_input	1	Replacement for real interpolated input concentrations for tracer 1 (same for tracer 2 .. 9), only with an according counter before _input
drainage_depth	1	grid with depth of drainage pipes in the soil; → mandatory when using the drainage option in [unsatzon_model]
drainage_distance	1	grid with distances of the drainage pipes or hoses from each other in m; → mandatory when using the drainage option in [unsatzon_model]
clay_depth	1	grid with the depth of an unpermeable layer (0 if no clay layer exists) → mandatory if swithcing on this feature in [unsatzon_model]
max_ponding_storage	1	grid with height of small dams around the fields for water ponding (in m) = maximum lake depth. 0 if no ponding occurs. For the lake model, this value is the theoretical value when the pond overflows.
irrigation_codes	0	Codes for irrigation referring to the irrigation table
lake_codes	0	Codes for each separate lake; mandatory when using the lake model
CriticalShearStress	1	Critical shear stress for mud avalanches; for Topmodel-version only
root_cohesion	1	root cohesion for soil stability index [N/m^2]
soil_cohesion	1	soil cohesion for soil stability index [N/m^2]
wet_soil_density	1	wet soil density for soil stability index [kg/m^3]
internal_friction_angle	1	soil internal friction angle for soil stability index as [-], not in 0..90°!
soil_depth	1	total soil depth (or at least depth to the instable layer) for soil stability index [m]
albedo	1	Reflectivity (albedo); it is recommended to use

Internal identification	Recommen-ded fillcode	comment
		this grid with one of the additional options for writing or to place it in the section variable_grids
Leaf_area_index1**	1	Leaf area index; usefull when using extended options (like writing statistics)
RoughnessLength1**	1	Roughness length; usefull when using extended options (like writing statistics)
root_depth1**	1	Root depth in m; usefull when using extended options (like writing statistics)
vegetation_coverage_degree1**	1	Vegetated fraction of the soil; useful when using extended options (like writing statistics)
SurfaceEvaporationResistance**	1	Surface resistance of the soil; useful when using extended options (like writing statistics)
SurfaceIntercepResistance1t**	1	Surface resistance fo the interception storage; usefull when using extended options (like writing statistics)
SurfaceCanopyResistance1**	1	Surface resistance of the vegetation; usefull when using extended options (like writing statistics)
ThawCoeffPermaFrost	0	grid with coefficients for a simple permafrost thawing model (nodata if no permafrost soil is present, else a suitable alpha value) → only for the deprecated permafrost model, not for the physically based heat transfer model.

* : when using multiple aquifers, each aquifer need such grids using appendix 2, 3 etc. instead of 1.
** : for each vegetation layer such a grid can optionally be provided (using the layer code instead of 1)

4.8.1.2 Options for standard grids (periodically reading, writing, statistics etc.)

There are a number of options which can be used to modify the behavior of a standard grid. All of the optional parameters start with an identifier followed by an equal sign and by the parameter value (or values). If an optional parameter is used, the equal sign and all the required (sub-)parameter values must be provided as well.

Here is an example of a standard grid definition using all possible parameters:

```
#$inpath//$laigrid leaf_area_index1 periodicity = 1 D 12 persistent = 0 12
fillcode = 1 defaultValue = 5 writecode = 2 readcode = 0 outname =
$outpath//$laigrid statfile = $outpath//$laistat statcode = 2001
```

(leaf area index is read in as a standard grid → this overwrites the internal leaf area index calculation; each day at 12 noon the grid is read in (following the naming scheme as defined below), the newly read in grid is not persistent but valid for 12 hours only. Other parameters are self explaining. This example doesn't make sense, it's constructed for explanation only.

- periodicity = <periods counter> <period> <sub-periods> : periodical re-reading of standard grids during the model run.
  <periods counter>: the number of periods to wait until a new grid is read in,
  <period> period (H=hour, D=day, M=month, Y=year)

<sub-period> number of sub intervals of the <period>.
Periodicity 2 D 12 means, that each other day (second day, i.e. 2 days, where 2 is <periods counter>) at hour 12 (12 = <sub-periods>) a new grid is read in. The grids must have a file name matching the original file name of the standard grid extended by the date-time-scheme. in the above example, the grid name may be m500.bh1, so the next standard grids must have the names:
m500.bh1_1984010212
m500.bh1_1984010412
m500.bh1_1984010612
m500.bh1_1984010812
etc. (scheme: filename.ext_yyyymmddhh)
This feature will be interesting for reading known changing boundary conditions for the GW-model as well as for changing land use in long term simulations. The only prerequisite is the creation of all the needed grids in advance to the model run. If a grid is missing, the model run will stop with an exception (and no results are written, so be careful to prepare all the required grids)

- persistent = <0|1> <replaceAfter> if the new read in grid should be persistents (default=1) or not. <replaceAfter> defines the number of timesteps when the newly read in grid is replaced by the original grid again. Using this option it is possible to define e.g. a boundary condition which is valid for a few month only and then be replaced by the original boundary condition grid.
- fillcode = <fillcode> (no default) either fillcode = <code> is used, or the code must appear as third parameter without the fillcode-keyword (as in older versions for compatibility). possible values: 0 = do not fill any nodata values; 1 = nearest neighbor; 2 = fill nodata with default value. If readcode = 0, the default value is used for the entire grid (because there is no nearest neighbor). If readcode = 1, then fillcode must be 2 in order to replace nodata values with the default value. If fillcode = 1 (and readcode = 1), then the old behavior of using the nearest neighbors value to fill missing cells is used.
- defaultValue = <defaultValue> (default: 0) used to fill missing values in the input grid, if readcode = 1 and fillcode = 2; used also for initializing internally created grids(if readcode = 0)
- writecode = <writecode> (default: 0) use one of the common codes for writing grids, e.g. 5 = per time step one grid, 55 = one average grid at the model end plus one grid per time step etc. As a special code, writecode 8 is for writing the grid after reading it in and filling missing values or using defaultValue to fill the cells. This can be used to create standard grids very easily. The filled grids are written to the defaultOutputDirectory with the original gridname and the appendix .filled.
- readcode = <readcode> (default: 1) defines, if the grid should be read in (1) or internally created (0) . if the parameter is ommitted, the old bahaviour is choosen (readcode=1)
- outname = <outname> (empty by default) defines the name or base name for the output grid. If no path is given, the value of the parameter DefaultOutputDirectory will be used as path
- statfile = <statisticfile> (no default) defines the name of an output statistics file.
If no path is given, the value of the parameter DefaultOutputDirectory will be used as path
- statcode = <statcode> (default: 0) example: statcode = 2001 defines the output configuration of the statistics file as for all other statistics (mean values or sums over nnn time steps for the entire basin plus optionally all sub basins or for all elements n the [output_list])

4.8.2 variable grids

Variable grids may also (exclusive or!) be defined in the section [standard_grids]. However, most control files place grids which will either be used in multiple sub models or which will be changed and written back to disk in this section. A list of possible grid with their identification strings is given in the following table

Table 4.8.2: list of variable_grids identification codes

Internal identification	**Recommended fillcode**	**Recommended default value for emtpy cells**	**comment**
GlacierizedCells	0	-9999	glacierized fraction of each cell (0...1, -9999 for all-time non-glacierized cells) when using the dynamic glacier model; WaSiM will check if there are only nodata cells. If yes, the _ice_firn_ grid will be used for initialization of the glacier cells; readgrids should always be 1 since otherwise no glacier would be created
GlacierCodes	0	-9999	codes for each single glacier. This grid is required when using the dynamic glacier model. It separates multiple glaciers even in the same subbasin for a applying the V-A-relation correctly; readgrids should always be 1 since otherwise no glacier zones could be created in the dynamic glacier model
soil_storage	1	0	soil water storage (Topmodel only); readgrids 0, if soil_storage should be derived from soil types, 1, if it should be read from file
soil_stability_index	1	0	soil stability index result.
SurfStorSiltingUp	1	1	storage for surface runoff which was routed into other grid cells but not into a cell with a river; used in Richards version only; required if surface routing or siltingUp model are enabled.
ETR2ETP	1	1	effectice for wasim-richards only: ETR/ETP fraction, used for dynamic irrigation amount modelling in irrigation method 4
ponding_storage_top	0	0	effectice for wasimtop only: pond grid for lake modelling, nodata values grid must not be filled; readgrids: 0, if ponds should be initialized in routing model by the volume-waterlevel relation, 1 if actual pond content should be read in from existing pond grid
VegetationStart1**	0	-1	JD for vegetation period start (optional when results should be stored)
VegetationStop1**	0	-1	JD for vegetation period end (optional when results should be stored)
VegetationDuration1 **	0	-1	Daycount for actual vegetation period

Internal identification	Recommended fillcode	Recommended default value for emtpy cells	comment
			(optional when results should be stored)
SumOfForcingUnits1**	0	-1	Sum of forcing units until phenological cycle starts (optional when results should be stored)
Pheno_start1**	0	-1	actual starting day (JD) as calculated by forcing units sum (optional when results should be stored)
SumOfChillingUnits1**	0	-1	Sum of chilling units until DP2_t1_dorm is reached -> FStar is calculated dependent on this values (optional when results should be stored)
FStar_ForcingThreshold1**	0	-1	FStar value to be reached by the sum of forcing units until dynamic phenology starts (optional when results should be stored)
SnowFreeDaysGrid	0	0	grid with number of effective snow-free days for permafrost soils (even if there is snow, snow free days will be reset only after a certain number of snow cover days is reached) → to be used with the deprecated permafrost model only
SnowCoverDaysGrid	0	0	grid with number of snow cover days, see above for more information; → to be used with the deprecated permafrost model only

** : for each vegetation layer such a grid can optionally be provided (using the layer code instead of 1)

4.9 Defining outputs

During the model run a number of output files are created controlled by the specifications in the control file. Possible outputs are either grids describing the spatial distribution of fluxes or states accumulated/averaged over a given time period or statistical output files containing spatially averaged values of fluxes, storage contents or states as time series. For both categories, control mechanisms are available controlling if, how often and what type of result should be written to disk.

4.9.1 statistics output

Generally most of the data which can be stored as grids also can be stored as statistic files. The statistic files are very useful for calibrating and validating the model. They can easily be imported into any graphics software to draw time series of modeled fluxes and states. The file format is similar to the format of the meteorological input data. Columns are separated by tab stops. The first row contains the zone number the column is valid for, the second row contains a data description,

and the third row contains the relative fractions of each subbasin/zone on the entire watershed area. Depending on the write code for the statistic file there will be also an additional last column which contains the average values for the entire model domain (except routed runoff). Using the programs RESAGG, RESAGGR, RESMEAN and RESEMANI the statistic files may be aggregated in order to get balances for month's and years (e.g. aggregated sum of evaporation, runoff and precipitation). In combination with the initial and final storage content values it is possible to check the balance of the model.

example of a statistic result file:

```
YY MM DD HH 1.00   2.00   3.00   4.00   5.00   6.00   7.00   8.00   9.00   10.00  11.00  12.00  tot_average
temperature_interpolted_using_regression_from_temper84.out,_zones_from_m500.ezg
-- -- -- -- 0.238  0.079  0.042  0.185  0.075  0.035  0.043  0.009  0.186  0.002  0.053  0.047  1.0000
84  1  1  1 1.993  2.766  3.274  3.154  4.934  5.010  4.140  4.849  4.669  4.748  5.138  3.442  3.516
84  1  1  2 1.832  2.794  3.422  3.298  5.056  4.750  3.951  5.326  4.738  5.282  5.142  3.198  3.512
84  1  1  3 1.650  2.862  3.620  3.474  5.320  4.818  4.047  5.744  5.007  5.834  5.340  3.280  3.611
84  1  1  4 1.278  2.252  2.906  2.851  5.215  5.550  4.768  5.022  4.898  4.778  5.484  4.178  3.398
84  1  1  5 0.632  1.286  1.727  1.699  3.461  4.729  4.410  3.131  3.673  2.963  4.027  4.526  2.414
84  1  1  6 0.604  1.577  2.241  2.242  4.966  6.215  5.557  4.475  4.859  4.153  5.519  5.274  3.121
84  1  1  7 0.325  0.857  1.214  1.189  2.622  3.700  3.623  2.388  2.819  2.256  3.068  3.936  1.809
```

Which results for what subbasins are written in which temporal accumulation is defined by a write code following the file name in the control file. The write code has 2 parts. The first digit controls the type of aggregation and the selection of considered zones. The second part (three digits) defines the temporal aggregation.

explanation of the codes:

0...1000 no output

1<nnn> spatial average *only for the entire basin,* temporally averaged over <nnn> time steps

2<nnn> spatial average for all zones *and* for the entire basin, temporally averaged over <nnn> time steps

3<nnn> spatial average *only for the entire basin,* temporally summed up for <nnn> time steps

4<nnn> spatial average for all zones *and* for the entire basin, temporally summed up for <nnn> time steps

5<nnn> spatial average only for the zones listed in the output list section of the control file and for the entire basin, temporally averaged over <nnn> time steps

6<nnn> spatial average only for the zones listed in the output list section of the control file and for the entire basin, temporally summed up over <nnn> time steps

examples:

2001 one output value (spatially averaged value) per time step for all zones and for the entire model domain,

2004 spatially averaged outputs for all zones and for the entire model domain, temporally averaged over 4 model time steps (e.g. output at hours 4, 8, 12 etc. if running in hourly time step)

4024 spatially averaged outputs for all zones and for the entire model domain, temporally summed up over 24 time steps (e.g. daily evaporation if running in an hourly time step)

5012 spatially averaged output only for the zones listed in the output list section of the control file and for the entire model domain, temporally averaged over 12 time steps

3120 spatially averaged output only for the entire basin, temporally summed up over 120 time steps

Especially if there are many zones or subbasins coded in the zone grid the size of output data would be quite large and not easy to analyze. If in a model run with a few hundred zones all files would be written using the code 2001 for a model year of 8760 time steps (hourly resolution), an amount of some hundred Megabytes would be stored to the disk! Therefore it may be better to store only those results in such high resolutions which are really important for the post processing. Also the selection of a reasonable output list (using e.g. only 20 of the 100 zones, defined in the section

[output_list]) will reduce the amount of data considerably.

4.9.2 Grid output

All grids that will be written by WaSiM are written by default in single precision binary format. It is possible and recommended to define the required default format in section [output_interval] of the control file by some optional attributes:

WriteAsciiGrids = 0 → all grids are written in ASCII format (ready to be read in by e.g. GIS software)

WriteDoubleGrids = 0 → all grids are written in double precision (internally, all grids are stored in double precision anyway and for some entities like groundwater pressure heads it is quite important to have that precision, so it is recommended to use double precision if possible). WriteDoublePrecision is not regarded if WriteAsciiGrids is active.

WriteNetcdfGrids = 1 → all grids are written in netCDF files. Actually **(as of version 10.02.02, the conventional grids are written as well until later the export as netCDF can be used exclusively).** There is no special parameterization required for writing grids and stacks as netCDF files. It's all done automatically. Stacks are written as layers or time steps of a 2D-grid into a single netCDF file. For more information on how to use netCDF files in WaSiM see chapters 4.6.1 and 2.3.9 .

If and how often and in which accumulation a grid is stored to the disk is specified in the control file for each possible output grid. There is a write parameter for each grid consisting of up to four digits.

The first digit (most significant digit MSD) controls the output of vegetation period based accumulations (this is the "thousands" digit). The second digit ("hundreds") controls the output of minimum and maximum grids, the third digit ("tens") controls the output of accumulated or averaged grids and the last digit (lowest significant digit LSD) controls the output of actual storage or flux grids.

The names of special, i.e accumulated, output grids are constructed by taking the first 4 characters of the original name and appending the next characters depending on the type of output. For example, minimum or maximum grids are stored using the same name as the original grid but replacing the extension by ".min" or ".max". For vegetation period based accumulations, the grid names are extended with _vegper and the extension is either sum or mit (the latter stands for averages).

Possible values for the first of the four digit are (only if write code ≥1000: 1. digit (1nnn, or 2nnn))

0: no vegetation period based grid is written

1: sum grid is written for vegetation period (summing up each value as long as this cells vegetation period is active → useful in combination with dynamic phenology only)

2: average value grid is written for vegetation period (summing up each value as long as this cells vegetation period is active → useful in combination with dynamic phenology only)

Possible values for the second of the four digits are:

0: or no first digit at all: neither minimum nor maximum grid are written
1: a minimum grid will be written when the model finishes the run (for each grid cell the minimum value for this grid cell of all time steps of the model time is stored)
2: a maximum grid will be written when the model finishes the run (for each grid cell the

maximum value for this grid cell of all time steps of the model time is stored)

3: minimum and maximum grid are stored

The code for the third digit of the grid write code can be set to the following values. The name of the resulting grid is generated from the first four characters of the original grid name followed by 4 digits for year and month or month and day if using codes 2, 3, 6, and 7 or 4 and 8, respectively.

0 or no second digit (write code has only 1 digit): no sum and/or average grid is written

1 a sum grid with accumulated values over the entire model time is written at the end of the run

2 sum grids with accumulated values are stored for each model year at the end of the year

3 sum grids with accumulated values are stored for each model month at the end of the month

4 sum grids with accumulated values are stored for each model day at the end of the day

5 a grid with averaged values over the entire model time is written at the end of the run

6 grids with temporally averaged values are stored for each model year at the end of the year

7 grids with temporally averaged values are stored for each model month at the end of the month

8 grids with temporally averaged values are stored for each model day at the end of the day

The last digit of the grid write code which is responsible for the output of actual grids can have one of 6 possible values. If the grid code consists only of one digit, this digit is the parameter which controls the output of actual grids and no minimum, maximum, sum or average grids are written at all. Possible values are:

0 no output of actual grids

1 the grid is written in each time step, it always gets the same name, old grids are overwritten → useful for debugging only, slows down the model performance considerably!

Note: when using the option WriteNetcdfGrids = 1 in section [output_interval] of the control file, then all files will be written to a single netCDF file as separate layers. That can be quite interesting for generating time series of interpolated data for later use in another model application, since netCDF data use compression and thus these grids are less storage intensive than normal ASCII or binary grids.

2 the grid is stored with unique names for each time step. The grid name is generated using the original grid name and the actual model date (e.g. the precipitation for the 14th of January, 7:00 will get the name “prec0114.007” if the original grid name (from the control file) is e.g. “precm500.grd”)

3 only the last grid of the model run will be stored. This is the way to get initialization grids for following model runs. The grid name will be the same as the original grid name in the control file

4 a new grid is written after a the end of a model day. The grid has always the same name. This is similar to code 1, in fact if using a time step of 1 day it is the same, but if the time step is shorter, e.g. 1 hour, this reduces write access to the disk

5 like 4, but the grids are stored using unique names like for write code 2 (but only one grid per day); at model end, an additional grid with the normal name is written as well (as for code 3)

6 actual grid is written at the end of each month with a unique name; at model end,

an additional grid with the normal name is written as well (as for code 3)

7 actual grid is written at the end of each year with a unique name; at model end, an additional grid with the normal name is written as well (as for code 3)

8 write grid immediately after reading the grid from file and filling missing values. This is used for an automated filling of missing values only (in particular in section standard grids). At model end, an additional grid with the normal name is written as well (as for code 3)

If the write code would be 353, the following grids would be written to the disk:

- a minimum grid for the entire model run, e.g. “tempm500.min”, annual minimum temperature
- a maximum grid for the entire model run, e.g. “tempm500.max”, annual maximum temperature
- a grid containing temporally averaged values for the entire model period, e.g. “tempm500.mit”, annual mean air temperature
- a grid containing the last actual values, e.g. “tempm500.grd”, the actual value of the last model time step (e.g. of 31st of December if the model run stopped at the end of the year)

If the write code would be 352 instead of 353, the actual grids would be stored at each time step. The grid names would range from “temp0101.001” to “temp1231.024”. The disk space requirements are enormous, so this writecode is recommended only for short periods or when looking for errors in the data.

If the last digit of the writecode is not 1 or 3, then the last state of the model run is written anyway. So a writecode of xxx2 means, that the actual grid is written for each time step following a naming schema as described above and at the end of the model run this same grid (i.e. the last actual state) is then written again with the originally configured name (usually with the extension .grd), in order to have that grid available for initialization of subsequent model runs. That makes it easier to handle situations when some grids are required as grid series (write code 2) but at the same time the last state should be stored for re-initialization pruposes.

4.10 Parametrizations of land use and soil types

Land use table and soil table are parametrized in asimilar way. Although the old versions of both tables will probably be usable for some time it is highly recommended to use this new version of both tables only. This newer format offers much more flexibility and extensibility for future new parameters and it doesn't require such rigid format than the old table.

Both tables follow this basic format:

```
<ID> <identification> {
      method = <method>;
      <parameter(-set) 1>
      <parameter(-set) 2>
      …
      <parameter(-set) n>
}
```

Here, <ID> is an integer referring to the land use or soil type code from the [multilayer_landuse]-table or the soil types grid, respectively. The <identification> should be a single word (without spaces) to identify the soil type or land use to the user (not used internally). All parameters are defined between the brackets {}. A siongle parameter or a group of parameters always start with “<parameter name> =”, followed by the single parameter or the parameter groups. The end of each such definition is always a semi colon “;”.

4.10.1 Land use parameter (multilayer_landuse table and landuse_table)

Parametrizations of land uses is done using two different tables:

- the first table is the [multilayer_landuse] table, defining codes for each combination of single land uses which must match to the cells of the land use grid. Each single land use in such a combination is characterized by another code, referring to the [landuse_table]
- The second table, the [landuse_table], defines the parameters for each of the single land use codes referred to by the [multilayer_landuse] table.

4.10.1.1 Multi layer land use table

The table defined in the control file section [multilayer_landuse] defines combinations of basic land uses as listed in the following section.

Here is an example of that table:

```
[multilayer_landuse]
10 # count of multilayer landuses
1 water { Landuse_Layers = 1,    -9999, -9999;
          k_extinct = 0.3;
          LAI_scale = 20;
         }
2 settlements    { Landuse_Layers = 2, -9999, -9999; k_extinct = 0.3; LAI_scale = 20;}
3 pine_forest    { Landuse_Layers = 3,     8, -9999; k_extinct = 0.3; LAI_scale = 20;}
4 deciduous_for. { Landuse_Layers = 4,     8,     7; k_extinct = 0.3; LAI_scale = 20;}
5 mixed_forest   { Landuse_Layers = 5,     8,     7; k_extinct = 0.3; LAI_scale = 20;}
6 agriculture    { Landuse_Layers = 6, -9999, -9999; k_extinct = 0.3; LAI_scale = 20;}
7 grass_variable { Landuse_Layers = 7, -9999, -9999; k_extinct = 0.3; LAI_scale = 20;}
8 bushes         { Landuse_Layers = 8,     7, -9999; k_extinct = 0.3; LAI_scale = 20;}
15 rock          { Landuse_Layers = 15,-9999, -9999; k_extinct = 0.3; LAI_scale = 20;}
19 horticulture  { Landuse_Layers = 19,-9999, -9999; k_extinct = 0.3; LAI_scale = 20;}
```

As can be seen from the first entry, it is not required to put all parameters into one row. However, it is important to enclose each entry with brackets “{}” and to close each single parameter with a semi colon.

The <id> entries of that table refer to the cells in the land use grid. The name is for user convenience only.

***Note**: the land use type “water” is an example only. It should rather not be used, since water does need some special parameter handling. To apply land use water to some cells, consider to use the lake model (see chapters on lake model and evaporation).*

Table 4.10.1: Reference table of the parameters for an entry in the multilayer_landuse table

Parameter	**Sample values**	**comment**
Landuse_layers =	List of (single) land use codes like 4, 8, 7;	Each code in the list referres to an entry in the basic land use table. The example here stands for trees (basic land use code 4), bushes/shrubs (basic land use code 8) and herbs/grass (land use code 8); the first land use code is assumed to define the uppermost layer, the next one comes as second layer etc.
k_extinct =	0.3;	extinction coefficient of d'Lambert-Beer's law for

Parameter	Sample values	comment
		reducing radiation in its way through the canopies (after COUPmodel, Jansson and Karlberg)
LAI_scale =	20;	Scaling factor for calculating the aerodynamic resistencies of layer 2..n dependent on the cumulated leaf area index (after COUPmodel, Jansson and Karlberg)

4.10.1.2 Basic land use table

Here is a complete list of all possible entries in the parameter section of the land use table (compare also chapter 2.8.3). The <ID> code of such an entry is referred by the Landuse_layers list of the [mulitlayer_landuse] table.

table 4.10.2: Reference table for all possible parameters for a land use table entry

Parameter	Sample values	comment
Method =	VariableDayCount DynamicPhenology_1 DynamicPhenology_2 DynamicPhenology_3 DynamicPhenology_4	Mandatory; Fix sample days for all phases dynamic sample days, see the respective chapter 2.23 on dynamic phenology.
RootDist =	-1 … +1	Mandatory; Controlling the root density and it's change with depth
TreduWet =	0.9...1.0	Mandatory; relative Theta value for beginning water stress (under wet conditions -> set >= 1 for crop which doesn't depend on an aerobic zone
LimitReduWet =	0.5...1	Optional (default=0); minimum relative reduction factor of real transpiration when water content reaches saturation. The reduction factor value will go down linearly starting at 1.0 when relative Theta equals TReduWet (e.g. 0.95) to LimitReduWet when the soil is saturated (Theta rel = 1.0)
UseInternalParameters =	00000000 … 11111111 positions: 1 = Albedo, 2 =	each single position switches the internal estimation of time variant parameters ON, a 0 switches it OFF --> those parameters must be read from external grids (whereas the cells with a landuse with a 1 at this position will be

Parameter	Sample values	comment
	rs_evaporation, 3 = rsc, 4 = rs_interception, 5 = LAI, 6 = VCF, 7 = RootDepth, 8 = Z0	calculated as usual. The positions are in the order they are defined in the flags-string (see left)
HReduDry =	3.45 (3...5m)	Mandatory; hydraulic head (suction) for beginning dryness stress (for water content resulting in higher suctions, ETR will be reduced down to 0 at suction=150m)
StartVegetationPeriodForBalance =	Any sample day index, e.g. 2	Optional (default=0); the sampling point in the following JD-Table when the vegetation period starts, default = 0 (start of model run)
StopVegetationPeriodForBalance =	Any sample day index, e.g. 6	Optional (default=n+1) the sampling point in the following JD-Table when the vegetation period ends, default = n+1 (end of model run)
JDVegetationResetForBalance =	1...365	Optional (default=-1 = at model start); Julian day, when vegetation start and vegetation stop grids are re-initialized to -1 (northern hemisphere: usually day 1)
JDVegetationWriteForBalance =	1..365	Optional (default=0=at model end); Julian day, when vegetation period dependent grids should be written (usually just before JDVegetationResetForBalance, e.g. 365). Attention: this Value should be identical for all land uses, since grids cannot be written for specific land uses only, so a global variable should be used in the control file
IntercepCap =	0...1 mm	Optional (default=0): specific thickness of the water layer on the leafs in mm. if omitted here, the default parameter from interception_model is used
StressFactorDynPhen =	1...2	Optional (default=0); stress factor for slowing down the phenologic development when plants are under dry stress.
F* =	e.g. 175 (for DP1, only an example)	Mandatory for methods DP1, DP2 and DP4, else ignored; Forcing threshold to start the new phenologic phase
DP1_t1_dorm =	60 (end of February, example only)	Mandatory for methods DP1, DP2 and DP4, else ignored; starting day (Julian day number), forcing units will be summed up after this day of year (when using DP1 or DP3 or when falling back to DP1 from DP2 and DP4 in the first model year)
DP1_T_Bf =	0	Mandatory for methods DP1, DP2 and DP4,

Parameter	Sample values	comment
		else ignored; threshold temperature for a positive forcing unit after thermal time model (when using DP1 or DP3 or when falling back to DP1 from DP2 and DP4 in the first model year)
JDReset_TStart =	1	Mandatory for DP1 to DP4; Julian Day when TStart is reset to -1 and Forcing untis are reset to 0 for a new vegetation period
maxStartJDforDP1 =	150	latest start day for the model run to use DynamicPhenology_1. If start date is after this date, then TStart is set to maxStartJDforDP1 minus the delta of the next column (e.g. 150 - 18 = 132), so we assume that this start date meets a fully developed vegetation. If start day is even after DP2_t0_dorm, then the next year will use DP1 only
DP2_t0_dorm =	244	used for DynamicPhenology_2 only: starting day (julian day number), chilling units will be summed up after this day of year until DP2 _t1_dorm is reached
DP2_t1_dorm =	110	used for DynamicPhenology_2 only: starting day (julian day number), forcing units will be summed up after this day of year
DP2_T_Bf =	0	used for DynamicPhenology_2 only: threshold temperatur for a positive forcing unit
DP2_T_Bc =	11.1	used for DynamicPhenology_2 only: threshold temperatur for a chilling unit
DP2_Par_a =	303.2	used for DynamicPhenology_2 only: Parameter a in F*=a*exp(bC*)
DP2_Par_b =	-0.019	used for DynamicPhenology_2 only: Parameter b in F*=a*exp(bC*)
DP2_Offset_1 =	-3.4	used for DynamicPhenology_2 only: value for z1 in R_c(T_i)=(T_i-z1)/(T_Bc-z1) when z1 < T_i < T_Bc
DP2_Offset_2 =	10.4	used for DynamicPhenology_2 only: value for z2 in R_c(T_i)=(T_i-z2)/(T_Bc-z2) when T_Bc < T_i < z2
DP4_t0_dorm =	e.g. 304	Used by DP4 only: starting day (julian day number), chilling units will be summed up after this day of year
DP4_t1_dorm =	e.g. 32	Used by DP4 only: starting day (julian day number), forcing units will be summed up after this day of year
DP4_T0 =	e.g. 9	Used by DP4 only: threshold temperatur for chilling units (if T < T0)

Parameter	Sample values	comment
DP4_T1 =	e.g. 4	Used by DP4 only: threshold temperatur for forcing units (T > T1);
DP4_Par_a =	e.g. 1748	Used by DP4 only: tree specific parameter, see chapter on dynamic phenology 4
DP4_Par_b =	e.g. -317	Used by DP4 only: tree specific parameter, see chapter on dynamic phenology 4
DP4_T_xylstop =	e.g. 10	Used by DP4 only: threshold temperature, the moving average of temperature has to stay below for 5 days in order to start leave fall
DP4_t_xs1 =	e.g. 183	Used by DP4 only: time of year (julian day number), the moving average analysis starts from (for start leave fall estimation)
DP4_SPSLF =	e.g. 5	Used by DP4 only: (SamplePointStartLeaveFall) sample point ID which is replaced by the day of start leave fall with the original value as latest time
JulDays =	e.g. 1 -1 +10 258 288 319 349 ; for DP1 or 15 46 74 105 135 166 196 227 258 288 319 349 for VariableDayCount	Mandatory; Julian days for all following rows. Each parameter must match the number of Julian days given here! The count of days doesn't matter (these are the so called “sample days”)
(max) JulDays =		Same as JulDays, (max) is ignored for all methods except DP3
Albedo =	e.g. 0.25 0.23 0.22 … 0.32	Mandatory; List of values for each sampel day
Rsc =	e.g. 80 80 75 65 55 55 55 55 55 75 80 80	Mandatory; leaf surface resistance in s/m, one value for each sample point.
LAI =	Some LAI values	Mandatory; Leaf Area Index (1/1); one value per sample day
Z0 =	Some roughness length values	Mandatory; Roughness length in m; one value per sample day
VCF =	Some vcf values	Mandatory; Vegetation covered fraction; one value per sample day
RootDepth =	0...2m	Mandatory; Root depth in m; one value per sample day

Parameter	Sample values	comment
AltDep =	-0.025 … +0.025	Mandatory; altitude correction: positive values: JD for start of this phenologic phase is later, depending on elevation; negative value: phenophase starts earlier; one value per sample day
rs_interception =	Usually lower values than rsc	Optional (default 20) when using interception method 2 ; INTERCEPTION surface resistance in s/m; one value per sample day
rs_evaporatio =	e.g. 150	Optional (default 200); SOIL surface resistance in s/m (for evaporation only); one value per sample day
SoilTillage =	e.g. 90 250;	optional set of 1..n Julian days, defining days with soil tillage. Important for silting up model only (else ignored)
SR_cultivation =	conserving, nonconserving, other	Used in surface routing only; type of cultivating the surface, see value list;
SR_StemDiameter =	e.g. 0.05 0.07 … 0.1 … 0.05 etc.	Used in surface routing only; default: 0.01: diameter of the stalks or effective diameter of crop/tree stems (perpendicular to the streamlines); used for nonconserving land use (using a plough); one value per sample day
SR_ResidCovrg =	e.g. 0.2 0.25 0.3 ...	Used in surface routing only; default: 0.01 fraction of soil covered by vegetation, stones, and residuals (mulch); used for conserving land use only; one value per sample day
SR_CropDistX =	e.g. 0.3;	Used in surface routing only; default: 0.3 distance of crops in flow direction; used for non-conserving land use (using a plough) only; one value per sample day
SR_CropDistY =	e.g. 0.4;	Used in surface routing only; default: 0.3 distance of crops perpendicular to flow direction, used for non-conserving land use (using a plough) only; one value per sample day
SR_K_ST =	e.g. 10;	Used in surface routing only; default: 10 Manning-Strickler-roughness for the soil surface in (m^1/3)/s; used for non-conserving and other landuse only; one value per sample day
SR_C_WR =	e.g. 1.5;	Used in surface routing only; default: 1.5 form drag coefficient for a circular cylinder in a group; used for non-conserving landuse only; one value per sample day
SR_rough_coeff1 =	e.g. 0.00496 0.892 0.0311	Used in surface routing only; default: 0.0198 0.712 0.142; coefficients for #

Parameter	Sample values	comment
		k_st-calculation for conserving landuse, Reynolds number < 20000; three values for the three coefficients
SR_rough_coeff2 =	e.g. 0.00519 1.200 0.177;	Used in surface routing only; default: 0.0198 0.712 0.142; coefficients for k_st-calculation for conserving landuse, Reynolds number >= 20000; three values for the three coefficients
N-factor	e.g. 0.7 0.7 0.95 1.0 1.1 1.2 1.1 1.1 1.0 0.8 0.7 0.7	Optional (default: the values given in the heat transfer section for freezing and thawing conditions): Scaling factor to apply on the air temperature to get the upper boundary condition.

4.10.2 soil parameters (multi-horizon soil table)

As for the basic land use table, here is a complete list of all possible entries in the parameter section of the soil table. The <ID> code of an entry refers to the cell values of the soil grid.

table 4.10.3: Reference table for all possible parameters for a soil table entry

Parameter	Sample values	comment
Unsaturated zone related parameters (Richards-approach)		
method =	MulipleHorizons	Mandatory;The only permitted value; multiple soil horizons can be defined having different hydraulic properties each.
name =	Sandm1 Sandm2;	Optional (no default); short descriptions, one value per horizon
ThicknessScaling =	1.0;	General scaling for all numerical layers. This option can be used to vary the soil thickness in order to get higher or lower storage volume for groundwater. The total soil thickness is automatically set as thickness of the first aquifer. This feature is available starting with WaSiM-version 9.08.04
horizon =	1 2 ;	Optional (no default); ID for each horizon; one value per horizon, used for user convenience only.
layers =	e.g. 2 5 12	Mandatory; number of numerical layers for each horizon. The thickness of the layer is given by layers x thickness. All profiles must have an identical number of layers (for memory handling reasons only); one value per soil horizon
ksat =	8.25e-4 6.25e-4 ;	Mandatory; saturated hydraulic conductivity in m/s; one value per soil horizon
theta_sat =	0.43 0.40 ;	Mandatory; saturated water content (fillable

Parameter	Sample values	comment
		porosity in 1/1); one value per soil horizon
theta_res =	0.045 0.045 ;	Mandatory; residual water content (in 1/1, water content which cannot be extracted by transpiration, only by evaporation); one value per soil horizon
alpha =	14.5 14.5 ;	Mandatory; van Genuchten Parameter Alpha; one value per soil horizon
Par_n =	2.68 2.98 ;	Mandatory; van Genuchten Parameter n; one value per soil horizon
thickness =	0.16666 0.3333 ;	Mandatory; thickness of each single numerical layer in this horizon in m; one value per soil horizon
Par_tau =	0.5 0.5 ;	Optional (default=0.5); so called Mualem-parameter in van-Genuchtens parameterization; one value for each horizon
k_recession =	0.8 0.9 ;	Optional (default=1.0); ksat-recession with depth: factor of recession per meter (only applied for the uppermost 2 meters of the soil); one value per horizon.
maxratio	100;	Optional (default=1000); maximum quotient of effektive k-values between two layers is limited by this factor, using the resulting higher value as effective value for unsaturated zone water flow.
Par_m	0.6;	Optional (default: 1-1/Par_n): The van-Genuchten Parameter m. Be careful when using this parameter since the van-Genuchten equation may not work as expected!
EvapMaxDepth	0.15;	Optional (default: depth of first numeric soil layer): maximum depth for soil evaporation in m
Siltingup-model related parameters		
GrainSizeDist =	0.75 0.1 0.05 0.05 0.03 0.01 0.01;	Optional (defaults: `Sand=0.3333; Silt=0.3333; Clay=0.3334; Stones1=0.0; Stones2=0.0; Stones3=0.0; Stones4=0.0;`): when using silting up model, the grain size fractions for sand, silt, clay, and Stones1..4 must be given here. Stones1 = 2-6.3mm, Stones2=6.3-20mm, Stones3=20-63mm, Stones4=63-200mm. All grain size fractions must be ≥ 0 (sand must be >0); The sum of all fractions (sand, silt, clay, stones1..4) must be 1.0.
SU_Par01 =	65.0;	Optional; only mandatory for SiltingUp-method 3 (dynamic expressions)
SU_Par02 =	12.2;	Optional; only mandatory for SiltingUp-

Parameter	Sample values	comment
		method 3 (dynamic expressions)
SU_Par03 =	0.52;	Optional; only mandatory for SiltingUp-method 3 (dynamic expressions)
SU_Par04 =	-0.64;	Optional; only mandatory for SiltingUp-method 3 (dynamic expressions)
SU_Par05 =	0.013;	Optional; only mandatory for SiltingUp-method 3 (dynamic expressions)
SU_Par06 =	-1.03;	Optional; only mandatory for SiltingUp-method 3 (dynamic expressions)
SU_Par07 =	0.7;	Optional; only mandatory for SiltingUp-method 3 (dynamic expressions)
SU_Par08 =	-0.19;	Optional; only mandatory for SiltingUp-method 3 (dynamic expressions)
PMacroThresh =	10;	Optional (default=1000); precipitation intensity thresholding macropore runoff in mm per hour (not in m/s, because it's more convenient than to write it down in m/s, e.g. 5mm/h = 1.38e-6)
Macropore infiltration related parameters		
MacroCapacity =	5;	Optional (default=0); capacity of the macropores in mm per hour
CapacityRedu =	0.9;	Optional (default=1); reduction of the macropore capacity with depth -> pores become less dense. This Factor describes the reduction ratio per meter
MacroDepth =	1.5;	Optional (default=1); maximum depth of the macropores in meter.
Heat transfer model related parameters		
DryHeatCapacity =	800; or 400 850 800;	Optional (default defined in section [heat_transfer]); dry heat capacity in J/(Kg*K). Can be provided as one single value as bulk property for the entire soil or as single values for each soil layer
DryDensity =	1500; or 900 1450 1500;	Optional (default defined in section [heat_transfer]); dry density in m^3/m^3 ; Can be provided as one single value as bulk property for the entire soil or as single values for each soil layer
DryThermalConduct =	0.58; or 0.2 0.58 0.56;	Optional (default defined in section [heat_transfer]); dry thermal conductivity in W/(m*K) (or J/(m*s*K) → 1J = 1Ws). Can be provided as one single value as bulk property for the entire soil or as single values for each soil layer

Parameter	Sample values	comment
KMinFrozenSoil =	1e-12;	Optional (default defined in section [heat_transfer]); minimum hydraulic conductivity in m/s when the soil is completely frozen
Topmodel related parameters		
Fcap =	27.65;	optional parameter which is needed for Topmodel only: water content at field capacity in mm (default if omitted: 25)
MSB =	29.0;	optional parameter which is needed for Topmodel only: maximum soil storage in mm; (default if omitted: 35)
ksat_topmodel =	5.56E-8;	optional parameter which is needed for Topmodel only: Saturated hydraulic conductivity for the Topmodel approach; (default if omitted: 1e-5 m/s)
Suction =	290;	optional parameter which is needed for Topmodel only: suction at saturation front; (default if omitted: 300 mm)

Note: all entries in the soil table must have the same sum of layers, since the soil is stored in a 3D-Array with a fix number of levels (which are the numerical layers for all soil horizons)

4.11 Using dynamic time steps in several modules

Several modules of WaSiM have an autmatic time step optimization algorithms implemented in order to keep the model stable or to deliver more feasible results. Most but not all of the sub time step mechnisms can be controlled by control file parameters.

4.11.1 radiation correction

Since radiation correction is non linear when dealing with longer time steps (> 1 hour), the time ste can be split into sub time steps within the radiation correction module. The last parameter in section [radiation_correction] defines the number of sub time steps, the radiation correction uses. For each sub time step, the solar geometry is computed and the correction factor is estimated. The total correction factor is the radiation weighted average of the single correction factors (see description of radiation correction).

4.11.2 evaporation/transpiration

For potential evapotranspiration, long time steps (e.g. 1 day), but also hourly time steps (when sunset or sunrise falls within this time step) are split into a night part and a day part. For each part, the potential evapotranspiration is calculated separately and the result is then the sum of both parts. There is no option to switch this behavior on or off.

4.11.3 soil model

The implementation of the dynamic time step controller into the Richards model of WaSiM allows a more flexible and process-oriented soil parametrization as well as an improved simulation quality of vertical water fluxes in layered soils. Using this controller soil water fluxes can be simulated with a

very high level of numerical stability even in cases with extreme parameter setups (e.g. very thin soil layers combined with the pronounced hydraulic conductivity of sandy soils).

The introduced dynamic time step control algorithm (available in WaSiM since version 7.9) considers explicitly the Courant stability criterion in order to prevent possibly oscillations caused by the discretization scheme of the actual soil profile. The Courant criterion is defined as follows:

$$Co = \left| \frac{\Delta t \cdot v}{\Delta x} \right| \leq 1 \qquad \textbf{(4.11.1)}$$

with Co Courant criterion
Δt time step [s]
v velocity [m/s]
Δx spatial distance between the nodes [m]

After its transformation the boundary condition for a stable numerical solution are given by:

$$\Delta x \geq \Delta t \cdot v \quad \text{resp.} \quad \Delta t \leq \frac{\Delta x}{v} \qquad \textbf{(4.11.2)}$$

with Δt simulation time step [s]
v hydraulic conductivity [m/s]
Δx thickness of the actual soil layer [m]

In WaSiM the algorithm of dynamic time step control has been a slightly adapted:

$$\Delta t = MIN_{i=1}^{k} \left(\frac{\Delta z_i}{k(\Theta)_{rel,i} \cdot k_{sat,i}} \right) \qquad \textbf{(4.11.3)}$$

with Δt minimum time for flowing of water through the soil layer at the actual water content [s]
i layer index (from 1..k)
k index of last soil layer above the groundwater level
Δz_i thickness of the actual soil layer i [m]
$k(\Theta)_{rel,i}$ relative moisture-dependent conductivity of layer i
$k_{sat,i}$ saturated hydraulic conductivity of layer i [m/s]

If the Courant time interval is shorter than the predefined model time step ($\Delta t < I$) then a certain split factor has to be calculated in order to reduce the length of the original time step:

$$f_{split} = \left\lceil \frac{I}{\Delta t} \right\rceil \qquad \textbf{(4.11.4)}$$

with f_{split} split factor with $f_{split} >= 1$
I length of the selected time step interval (as defined in control file) [s]
Δt result from Equation (9.9.3) [s]
$\lceil \rceil$ character for a rounded up integer

The internal reduction of the (original) model time step is done separately for each grid cell. For this purpose the implemented control algorithm checks all soil layers above the groundwater level with regard to the actual Courant criterion. If necessary ($\Delta t < I$) then the (original) model time step is subdivided into a certain number of subintervals (f_{split}). In this case, the absolute input data [in mm] for the soil model have to be recalculated for each subinterval whereas data of flow velocity (e.g. descent and ascent rates of the groundwater level or water flows between soil layers) can be used as

before (i.e. without subdivision) because of their independence of the length of calculation time interval.

The shortest possible subinterval is one second. That means in effect that a model time step of one hour can be subdivided into up to 3600 subintervals. In order to increase the flexibility of the implemented algorithm, it is possible to define a minimum subinterval length [in s]. This is the only parameter for controlling the sub time step available in the unsaturated zone model.

The dynamic time step control can be activated by selecting method "3" in the section [unsatzon_model] of the WaSiM control file.

4.11.4 surface routing

For the surface routing it is crucial to set the sub time steps to a value range which allows the model to find an optimal value. Since the surface flow is highly dynamic, it may be necessary to set sub time steps of a few seconds, especially when using high grid resolutions.

There are two parameters controlling the sub time step of the surface routing:

- shortest sub-time step in seconds
- longest allowed sub time step (even if flow travel times are longer, the time step is subdivided into sub time steps of this length) be careful: tracers are mixed much faster when multiple sub time steps are applied

4.11.5 Groundwater

There is no internal sub time step controlling for the groundwater model. However, there is an externally controlled loop for the sub model group with the following sub models:

- surface routing
- unsaturated zone
- accumulation of real evaporation
- groundwater

The sub time step for this components can be set in the control file in a separate section called [soil_surface_groundwater_substeps].

The single one parameter allowed there controlles the number of sub time steps for the mentioned sub-model group. This may lead to better model results when using long time steps but dealing with dynamic coupling between surface and unsaturated zone and groundwater. The default is 1 (no sub time step splitting)

4.11.6 discharge routing

The discharge routing uses an internally optimized sub-channel length, which depends on the sub time step defined in the control file. There, the number of sub time steps must be defined as fix value <n>. Each routing channel is then routed <n> times (but with $1/n^{th}$ of a time step length).

4.11.7 lake model

Abstractions and volume balances of lakes and reservoirs can be controlled by a maximum time step defined in each reservoir definition (SP element) in a routing description. If a dTmin parameter is given there, then this parameter defines the sub time step to be used for all balancing in minutes

```
TG 10 (AE=135.250, AErel=1.0)
 from OL 233 (kh=0.1, kv=0.4, Bh= 2.0, Bv=  8.0, Th= 0.42, Mh=25.0, Mv=10.0,
```

```
I=0.0449, L=20071.0, AE=10.000)
 and SP 15   ( file = $outpath//spv___15_010.//$code//$year , V0 = 2.0E5, C0 =
0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0, dTmin = 60, UseRunningAvg = 12 )
```

4.12 Coupling with external models

WaSiM allows coupling with external models. This functionallity was especially developed for coupling with groundwater models, in particular the groundwater model PCGEOFIM (**P**rogramsystem for **C**omputation of **GEOFI**ltration and geo**M**igration; see http://www.ibgw-leipzig.de/) but it quite easy to adapt this coupling module to other external models. The WaSiM coupling module can be run with or without activation of the internal groundwater module. In fact, it is even possible to run two (or multiple) WaSiM instances that do diffenrent tasks: one instance does the interpolation, the next instance reads all interpolation results from the first instance and does the rest of the model time step (although this is not really an efficient approach).

The data exchange between WaSiM and the external model can be carried out at the end of each simulation time step or at the end of a pre-defined number of simulation intervals. The parameterization scheme of the coupling module is shown below (section [ExtarnalCoupling] in the WaSiM control file, the line numbers shown before each line are not part of the control file but for reference to the explanations below):

```
00    [ExternalCoupling]
01    1
02    $exchngpath//do_interpolations.inf
03    100
04    D
05    1440
06    1
07    $exchngpath//gwtable.grd      GWTableExtern           1     0

08    #$exchngpath//bh.grd          gw_boundary_fix_h_1     0     0

09    2
10    $exchngpath//gwn.grd    groundwater_recharge    Mean
11    $exchngpath//balance.grd      Balance                 SumTotal_MY
12    2
13    $exchngpath//qdir.tab   direct_discharge        Sum
14    $exchngpath//qifl.tab   Interflow               Sum
15    $exchngpath//do_wasim.inf
16  Command=Continue;
```

The line-by-line entries have the following meaning:

Line

01 activate / deactivate the external coupling module; 0 = no coupling, 1 = coupling

02 path and name of semaphore (or synchronization) file provided by the external model. This file may contain a single line with a command transferred to WaSiM. The following commands are recognized:

Command=Continue; → WaSiM will continue the model run

Command=Fisish; → WaSiM will finish the model run after the actual time step. Same as pressing Ctrl+C and typing “1”

Command=StopWithoutSaving; → WaSiM will cancel the actual model run without saving any results. Same as Ctrl+C and typing “2”

Command=SaveAndContinue; → WaSiM will flush statistic files and continue with the model run. Same as Ctrl+C and typing “3”

03 wait interval for scanning the exchange directory for the new semaphore file [ms]. Recommended range: 10...100 ms

04 Coupling mode: I = each interval, H = each hour, D = each Day, M = each month, Y = each year. Attention: If the coupling mode is not each interval ("I"), then grids that will be transferred from or to another model should contain either average values or sums and the other model should be aware of this fact. Different accumulations for grids can be defined for each grid that is written by WaSiM, see line 10

05 time interval used by the external model [min]. This is important for importing an externally generated groundwater table only (which is a special case, since there are some internal grids that need to be adopted after importing an externally changed groundwater table)

06 number of grids provided by the external model; The files must be available for WaSiM once the semaphore file was written and WaSiM will now read these files. Each following row (1..n as defined by the parameter in this line 06) will contain specifications for one grid to be written (stacks can not be written yet).

07 a line describing a grid to be read in by WaSiM: The first parameter is the path and file name, the second parameter is the internal grid name, and the third parameter is the "fillMissing" parameter (0 = no fill, 1 = fill with nearest neighbors value). Finally, the fourth and last parameter defines the rename mode either as 0 (= delete) or 1 (=rename).

08 here labeled (#) as comment; additional files of the external model could be defined here for declaring them to WaSiM as grids to be looked for when synchronizing the models

09 number of grids written by WaSiM to the exchange directory before the semaphore file is written and the next synchronization can be done (by the external model)

10 a line describing a grid to be written by WaSiM . The first parameter is the path and name of the grid, here the groundwater recharge grid, the second parameter is the internal grid name (this name is an internal identifier, a list of all usable identifiers can be found at the very end of this manual); third parameter is the accumulation method if the exchange is not done every time step (Last, Mean, Sum, SumTotal[_x] with x as D, M or Y or a combination thereof, see below).

11 another line describing the second grid to be written by WaSiM (only if the number of grids is defined as 2 or greater). First parameter: path and name of the second grid, here the balance grid (Due to the exchange of groundwater tables WaSiM has to adapt the internally calculated soil water content to the new condition. The cumulative amount of adaptation water is balanced by the balance grid.), second parameter: internal grid name, third parameter: Accumulation mode (Last, Mean, Sum or SumTotal with one or more of the following codes appended: D = daily grids, M = monthly grids, Y = annual grids; other options e.g. MY = both monthly and annual grids will be written, the calling program should deal with that data – in this example, none of the balance grids are really needed by the external groundwater model for proper functionality. The balance grids are simply a check on how much water gets lost or will be gained by synchronizing the two models repetitively)

12 number of sub-basin correlated (zonal) statistics (average values for all cells of a sub-basin or zone) which should be written as table (in ASCII-Format)

13 first parameter: path and name of the file (here for direct flow per subbasin/zone [mm/Δt]); second parameter: internal entity name, third parameter: accumulation method (Sum, Mean, Last)

14 first parameter: path and name of the 2nd file (here for interflow per subbasin/zone [mm/Δt]); second parameter: internal file name, third parameter: accumulation method (Sum, Mean, Last)

15 path and name of semaphore file provided by WaSiM after all of the output above was written

16 content of the semaphore file written by WaSiM. It may contain any text, but there are some commands available that can be used to control the program flow of the other model. See description of line 02 above for details.

The exchange data containing statistic information per sub-basin are provided by WaSiM as Ascii

tables with the following format:

```
direct discharge [mm per Zone] (QD) unsatzon_model; 4 zones
1       0.128897
2       1.3298
3       4.5257
4       2.14858
average 3.11232
```

The “internal entity name” should be taken as a distinct sub-string from the normal statistics file, line 2. Such files start with e.g

```
YY    MM    DD    HH    "1"   "2"   "3"   tot_average
snow_melt_(effective_outflow_from_snow_cover)____submodel:snow_model
--    --    --    --    0.100 0.600 0.300 1.000
```

The chosen sub-string should be long enough to make the name unique, e.g. “snow_melt_(effective_outflow” would be sufficient, but “snow_melt” would not, since there is another statistics file containing the following header:

```
YY    MM    DD    HH    "1"   "2"   "3"   tot_average
snow_melt_only(effective_outflow_from_snow_cover)____submodel:snow_model
--    --    --    --    0.100 0.600 0.300 1.000
```

For the latter example, the identification string could be “snow_melt_only”, since this is already distinct from any other statistics file header.

In Version 10.02.00, the exchange-mechanism was re-activated again and extended by a few features:

- runs in MPI mode as well as in normal executables
- runs on Windows and Linux
- semaphore file may contain a specific command (see description of line 02 above)
- when finishing the model run, WaSiM always and automatically does two things:
  - the content of the semaphore file which WaSiM writes, will be changed to “Command=Finish;”. This will cause the external model to finish as well, as long as the external model can interpret such commands
  - an extra semaphore file will be written, which has the same name as the normal semaphore file but with the extension “_Done”. This file also contains the line “Command=Finish;”. It can be used to make sure that WaSiM has finished it’s model run.
- The Wait-mechanism is performance optimized: There exist some library functions that make the running program sleep for a given amount of time without using any CPU resources. However, since these functions are not defined in the C/C++ standard, and the old _sleep() function is deprecated, the new functions are slightly different for Windows and Linux. There is, however, no difference in parameterizing the wait interval. It is given in milliseconds in both systems. However, when running WaSiM in MPI mode with multiple instances, only rank 0 will wait for the semaphore file. So only this instance will sleep, while all other instances wait at the next barrier to be synchronized with instance 0 again. That leads to 100% CPU-usage of the respective cores of all other ranks while rank 0 is waiting in sleep mode for the semaphore file. A possibility to overcome this on a single processor machine is to use multiple OpenMP threads in a single WaSiM instance instead of multiple WaSiM instances with no OpenMP threads at all. Such, the CPU usage during waiting periods is almost zero for that single WaSiM instance and can use all CPU cores

when active. However, on a multi-CPU computer (cluster or super computer) this is not feasible and OpenMP can not be used there over multiple CPU's anyway (so the CPU's will be fully used by the MPI version of WaSiM even when waiting for the semaphore file).

4.13 Using variable cell sizes

WaSiM cannot only run with a fixed cellsize but allows also the use of variable cell sizes. However, not all model components are working in this mode (ask the author, if detailled info is required).

Using variable cell sizes includes the following steps:

- Relative cell area grid may be given (see Standard grids), if not it is generated from csx and csy. If these grids are not given, too, it is generated containing in all cells the value 1.0
- grid of cell size in x direction **csx:** It may be given together with the rca grid and the csy grid. It contains the absolute cell size in x direction in meter for each cell. In case of given csx and csy files, rca is checked - or generated, if not provided, using the default cell size given in the zone grid to calculate the relative area by (csx*csy)/cellsize
- grid of cell size in y direction **csy:** the same like for csx but in y-direction.
- grid with x coordiates of cells **cx**: This grid is generated during model initialization. It considers the different cell sizes, using the left and the bottom edge of the grid as orientation (assuming South-North-direction for the left edge and West-East-direction for the bottom edge)
- grid with y coordinates of cells **cy**: See above under cx-grid
- internal generation of these grids → if following grids are provided: **rca** and **csx** and **csy**, or **rca** and one of **csx** or **csy,** or only **rca** or only one of **csx** or **csy:** If only one of csx or csy is read in and no rca is given, then the relative cell size is estimated using the standard cell size and the value given in csx or csy, resp.
- changes in the control file: read in as standard grid: rca-grid and/or one of csx/csy or both.
- using rotated coordinates in meteorological interpolations. All coordinate dependent functions use the real coordinates.
- the control file should be changed in the following way (only those grids must appear as standard grids, which are really there, mostly recommended is the csx and csy - combination

```
[ standard_grids]
<n>       # number of standard grids
…
…
$inpath//$RelCellArea_grid RelCellArea   1 # grid with relative cell area
$inpath//$CellSizeX_grid CellSizeX 1 # grid with cellsize in x-direction
$inpath//$CellSizeY_grid CellSizeY 1 # grid with cellsize in y-direction
…
…
```

4.14 Using shorter timesteps than one hour (minute time steps)

4.14.1 Introduction

Although many WaSiM applications already used time steps of less than one hour, there were still some drawbacks. The input format of meteorologic data with columns for year, month, day and hour meant, that no information about the actual minute was provided. This caused some problems with actions that were done at specific model times only, like

- setting new land-use parameters at the end of a day (was done multiple times or never, depending on the offset of the hour value, i.e. if the hour value for the first hour of the day was given as 0 or 1)
- writing grids at the last time step of a day (every time step with hour 23 or 24 was considered to be the last time step – this was particularly annoying when sums or averages should be caclulated)
- writing output statistics with accumulated (mean or sum) values over n time steps
- radiation corrections
- glacier balancing
- etc.

To allow the user a consistent and backwards compatible handling of time steps of less than one hour, WaSiM 10.05.03 provides full capabilities for time steps down to one minute. Still, no shorter time steps than one minute are allowed. The main changes include:

- meteorologic input data for time steps less than one hour must be provided with an additional column for the actual minute (header: mm, zero-based)
- same for observed runoff: a new column with header „mm“ describes the minute-values
- the control file contains an explicitly named parameter timestep=<n> (in minutes) in section [output_interval]. This is a mandatory parameter in order for the model to recognize the shorter time step. For models with hourly time step, the parameter may be omitted.
- The first hour value for a day must always be 0, the last value 23, otherwise the minutes couldn't be added to the hour 24 (if hour value ranges from 1...24)
- minute values range from 0 to 59
- time steps must be integer divisors of 60: Allowed are 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30
- the model time in section [model_time] must contain the start and end dates resp. times with explicitly named parameters like startminute = 0, starthour = 0, startday=1 etc.
- that's all...

Parameters for sub-models don't need to be re-calibrated, except if the results show that it may appropriate to re-calibrate parameters e.g. for the soil (e.g. to better get infiltration excess).

All external help tools that convert or visualize time series, were adapted to the additional minute column as well. They recognize the mm-header column automatically:

- resaggr
- resmean
- qtospend
- graphlines
- not adapted: regr (creating regression parameters, which must still be done with hourly data in regress.exe, the resulting parameters can be manually transferred to the control file, where interpolation methods with regression will use these parameters in minute time steps as well)

4.14.2 Requirements in detail

Control file changes

- set the $time variable to the actual time step, e.g.:

```
$set $time = 10.0
```

- define additional variables for startminute and endminute, use 0-based minutes and hours

```
$set $startminute = 00
$set $endminute = 50
```

- add the new mandatory named parameter timestep to the section [output_interval]

```
[output_interval]
25   # increment of time steps until an output to the screen is done
1    # warning level for interpolation (no station within search radius)
1    # output of runoff results in 0=mm/time step, 1=m3/s
$time  # minutes from the hour-entry in the input data files until the end
timestep = $time # optional for hourly but mandatory for shorter time steps
WriteAsciiGrids = 0
WriteDoubleGrids = 1
InitialStateDirectory = $InitialStateDirectory
DefaultOutputDirectory = $DefaultOutputDirectory
```

- use named parameters for all start- and end-dates and -times:

```
[model_time]
startminute = $startminute
starthour   = $starthour
startday    = $startday
startmonth  = $startmonth
startyear   = $startyear
endminute = $endminute
endhour   = $endhour
endday    = $endday
endmonth  = $endmonth
endyear   = $endyear
```

- check the names for meteorologic input data and change them if required to the ones representing the e.g. 10-min-inputs
- same for observed runoff data
- check time variable abstraction rules from lakes or rivers that were defined for 1-based hourly time steps to make sure, that the starting date is in 0-based notation. 24 is still possible as end time – meaning that all hours till the end of day are active
- if comparisons between e.g. 1-hour resolution and 10-min resolution should be done, make sure to choose m^3/s as measure for runoff routing, since the comparison between mm/time step si not feasible otherwise (mm/h and mm/10min cannot really be compared)

input files

Input files for meteorologic and hydrologic input data must contain an additional column for the actual minute when using a time step of less than one hour (this was NOT the case in older setups and this is one thing that is NOT backwards compatible).

Sample input for traditional 1-hour data:

```
Data for Converted_All, Column T
YY    MM    DD    HH    1538   1075   1320   3580   950    1015   1015   1980
YY    MM    DD    HH    629564 601449 609350 641930 595686 595870 595870 668583
YY    MM    DD    HH    141084 143518 148975 155275 155358 155480 155480 158215
YY    MM    DD    HH    BLA    69228  ABO    JUN    MMZ    ZWE    4273   GRH
1990  1     1     1     -9999  -9999  -2.8   -11    -9999  -9999  -9999  -6.2
1990  1     1     2     -9999  -9999  -2.9   -10.8  -9999  -9999  -9999  -5.2
1990  1     1     3     -9999  -9999  -2.9   -10.7  -9999  -9999  -9999  -5
```

Sample input for new 10-min data (slightly other station selection used here, since not all stations provided 10-min data):

```
Data for Converted_All_10min, Column T
YY     MM     DD     HH      mm    850    1335   1035   803    1015   3040   1270   570
YY     MM     DD     HH      mm    618005 635550 674156 603157 626064 675400 657550 611200
YY     MM     DD     HH      mm    196237 186385 186097 184316 182126 180400 177890 177640
YY     MM     DD     HH      mm    69061  10237  ENG    69073  69329  TIT    4199   THU
2010   1      1      0       0     2.700  -9999  -0.700 2.300  -9999  -9.700 -9999  -9999
2010   1      1      0      10     2.500  -9999  -0.300 2.300  -9999  -9.700 -9999  -9999
2010   1      1      0      20     2.700  -9999  -0.300 2.400  -9999  -9.700 -9999  -9999
2010   1      1      0      30     2.800  -9999  -0.100 2.500  -9999  -9.900 -9999  -9999
2010   1      1      0      40     2.800  -9999  -0.300 2.600  -9999  -10.20 -9999  -9999
2010   1      1      0      50     2.800  -9999  -1.000 2.500  -9999  -10.10 -9999  -9999
2010   1      1      1       0     2.800  -9999  -1.000 2.400  -9999  -9.900 -9999  -9999
2010   1      1      1      10     2.900  -9999  -0.400 2.400  -9999  -9.700 -9999  -9999
2010   1      1      1      20     2.900  -9999  -1.000 2.300  -9999  -9.500 -9999  -9999
2010   1      1      1      30     3.100  -9999  -0.900 2.200  -9999  -9.700 -9999  -9999
2010   1      1      1      40     3.100  -9999  -0.700 2.100  -9999  -9.000 -9999  -9999
2010   1      1      1      50     2.900  -9999  -0.900 1.900  -9999  -9.000 -9999  -9999
2010   1      1      2       0     2.800  -9999  -0.200 1.800  -9999  -9.100 -9999  -9999
2010   1      1      2      10     2.800  -9999  -0.100 1.700  -9999  -9.300 -9999  -9999
2010   1      1      2      20     2.800  -9999  -1.100 1.600  -9999  -9.900 -9999  -9999
2010   1      1      2      30     2.800  -9999  -1.400 1.400  -9999  -10.00 -9999  -9999
2010   1      1      2      40     2.900  -9999  -0.400 1.300  -9999  -10.30 -9999  -9999
2010   1      1      2      50     2.800  -9999  -0.700 1.400  -9999  -10.50 -9999  -9999
```

Note:

1. the new minute column has the symbol ***mm*** in all rows of the header (lower case)
2. the hour data changed from 1-based values to 0-based values
3. the minute data in 10-min time steps is also 0-based and ragnes from 0 to 50 for the 6 time steps of 10 min within one hour

output files

Like the input files, also the output files contain a new ***mm***-column when using a time step of less than one hour.

```
YY     MM     DD     HH      mm    "1"    "2"    "3"    "4"    "5"    "6"    "7"    "8"
       ...
precipitation interpolated with elevation dependent regression + idw, all values take
--     --     --     --     --     0.0478 0.0026 0.0617 0.0167 0.0024 0.0089 0.013 0.003
2018    1     1      0       0     0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
2018    1     1      0      10     0.001  0.000  0.000  0.000  0.000  0.001  0.006  0.000
2018    1     1      0      20     0.020  0.009  0.003  0.001  0.005  0.019  0.029  0.003
2018    1     1      0      30     0.080  0.030  0.008  0.003  0.022  0.038  0.055  0.007
2018    1     1      0      40     0.088  0.034  0.009  0.004  0.025  0.045  0.065  0.009
2018    1     1      0      50     0.203  0.075  0.022  0.002  0.064  0.101  0.306  0.022
2018    1     1      1       0     0.340  0.102  0.034  0.008  0.075  0.105  0.200  0.027
2018    1     1      1      10     0.445  0.387  0.051  0.018  0.109  0.148  0.338  0.036
2018    1     1      1      20     0.435  0.155  0.057  0.019  0.119  0.146  0.238  0.043
2018    1     1      1      30     0.272  0.124  0.068  0.032  0.097  0.178  0.136  0.041
2018    1     1      1      40     0.219  0.114  0.102  0.021  0.141  0.217  0.356  0.115
2018    1     1      1      50     0.187  0.114  0.214  0.034  0.273  0.155  0.220  0.364
2018    1      1     2       0     0.275  0.267  0.278  0.079  0.266  0.258  0.371  0.383
2018    1      1     2      10     0.179  0.119  0.219  0.063  0.209  0.147  0.242  0.312
2018    1      1     2      20     0.172  0.121  0.331  0.161  0.309  0.232  0.221  0.375
2018    1      1     2      30     0.123  0.073  0.246  0.198  0.194  0.102  0.081  0.272
2018    1      1     2      40     0.070  0.024  0.209  0.171  0.164  0.132  0.273  0.238
2018    1      1     2      50     0.037  0.039  0.228  0.247  0.250  0.197  0.122  0.257
```

Note:

1. the additional header marker for the minute is ***mm*** (lower case)
2. hours and minutes are 0-based as the input is

4.14.3 Sample results

The following figures were created from two different model runs, one of them with a resolution of 1 hour, the other with a resolution of 10 minutes. Input data for precipitation, air temperature, wind speed and global radiation were available in 10 min resolution, whereas input data for relative sunshine duration and relative humidity were available as hourly values only – the latter two had to be converted by simply adding additional lines into the input files with identical values for all 10-minute time steps of an original 1-hour time step.

The main problem is, that there are not sufficient station data for 10-min precipitations available. The number of stations providing 10-min-data is much smaller than the number of „normal" stations providing 1h-data. So most of the figures show a mixed effect of different time steps and different station networks. Only figure 4.14.3.1 compares modelled runoffs, that were modelled with precipitation data from the same station network (the 1h precipitation data were accumulated from the 10-min-data). There are only small differences mainly during the snow melt season, which may be a result of the applied energy balance approach, which is reacting more nonlinear to 10-min input data vs. 1h input data than simple degree-day-approaches.

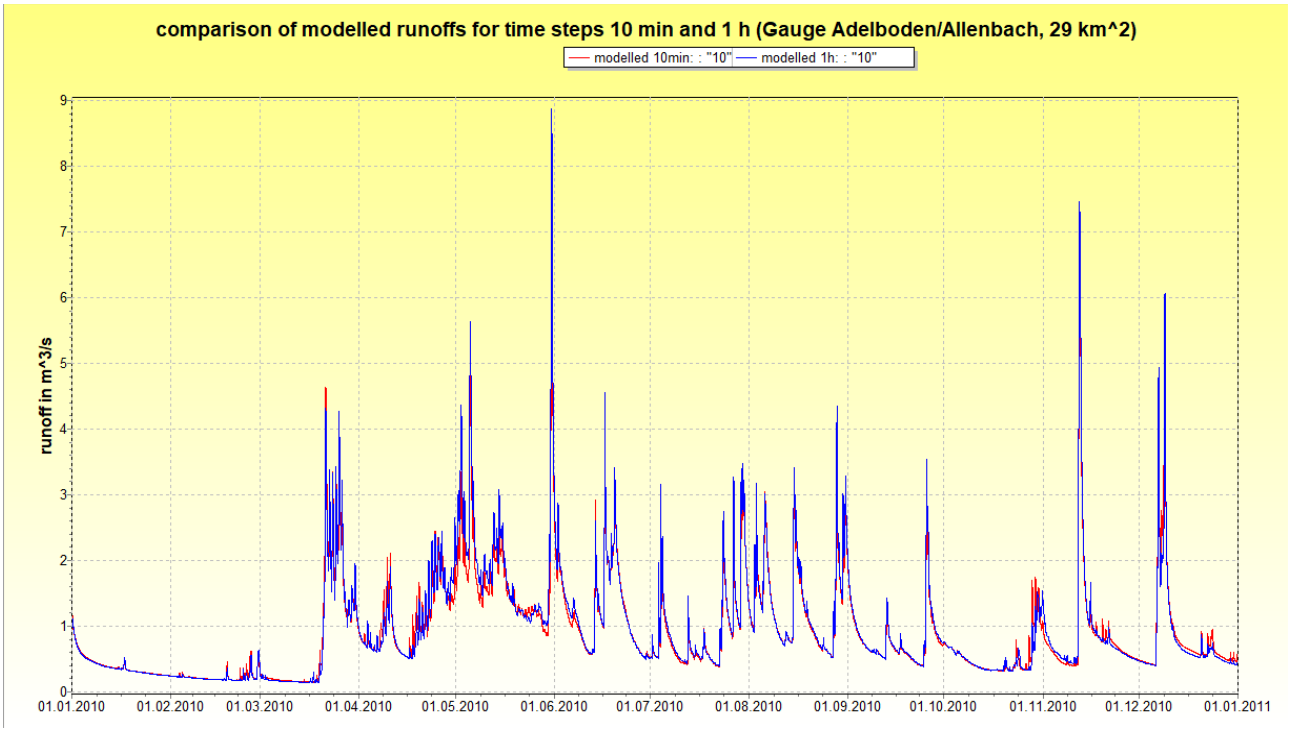


figure 4.14.3.1: comparison of modelled runoff for a 29 km2 subbasin of the river Kander (Canton of Berne, Switzerland) for 2010, red is for time step 10 minutes, blue for time step 1 hour, precipitation network is identical

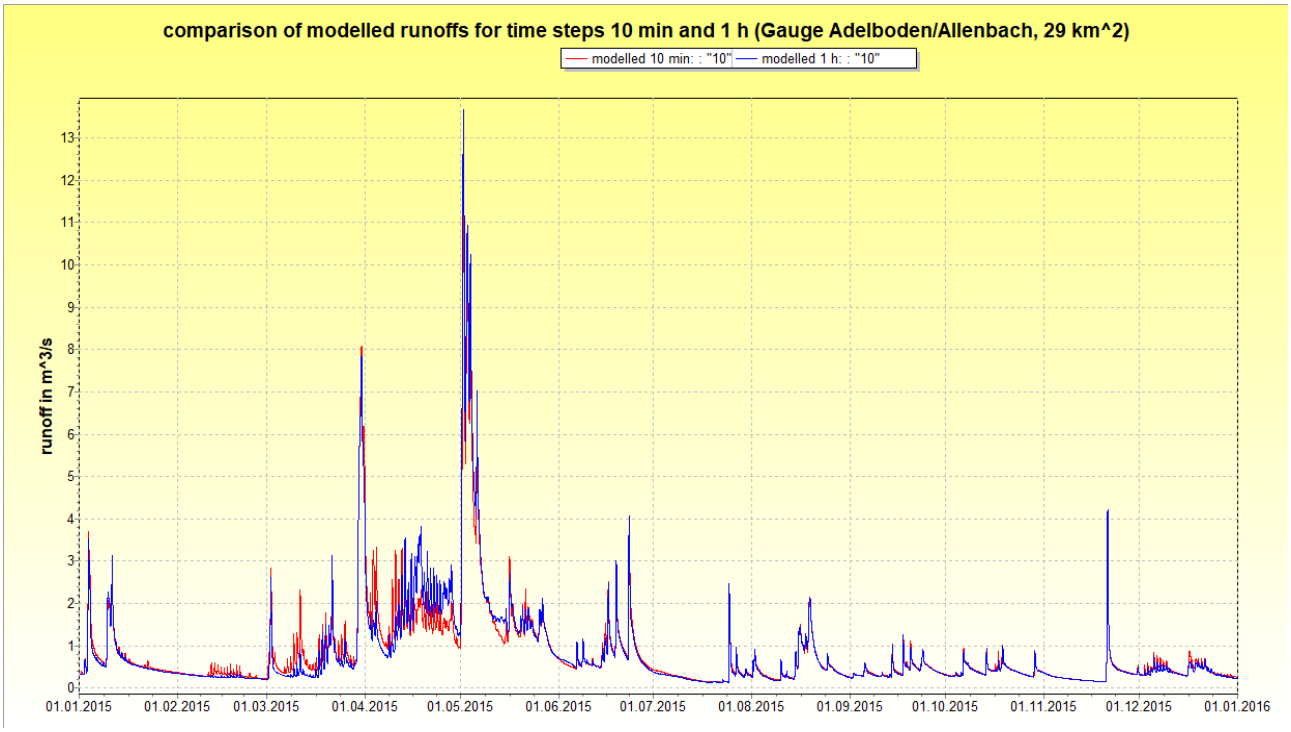


figure 4.14.3.2: comparison of modelled runoff for a 29 km2 subbasin of the river Kander (Canton of Berne, Switzerland) for 2015, red is for time step 10 minutes, blue for time step 1 hour

Figures 4.14.3.2, 4.14.3.3 and 4.14.3.4 show modelled and observed runoff (figure 4.14.3.2 only modelled runoff) for gauge Allensbach/Adelboden for 2015, zooming into the flood event of May, 1st from figure to figure.

Some things are quite obvious:

1. the overall runoff is (except for the melting period) almost identical for 10 min and 1 h time steps
2. during snow melt, differences appear larger. This is the effect of the above mentioned energy balance approach and probable non-linearities in this apporach.
3. The observed 10-min runoffs have quite a bit high frequency variations around the average value for 1 hour time steps
4. flood events in smaller basins tend to show higher peaks for 10 min time steps in observations (see also figures 4.14.3.5 and 4.14.3.6)
5. The modelled runoff behaves quite similar for 10-min time steps and 1 h time steps for larger sub-basins, in particular when there are no high resolution precipitation stations available in or nearby the modelled sub-basin.

Figure 4.14.3.4 shows that the modelled runoffs for 10-min time steps have a little bit more variability than the 1 hour time steps, although this is only a minimal effect. A real advantage will be there for very small basins only (e.g. <10 km ^{2}). This was not possible to test here, however.

Finally, figures 4.14.3.5 and 4.14.3.6 show two more examples for flood events in small basins that are quite different between 10-min and 1-hour resolution but that were both not modelled at all because no precipitation was interpolated (local convective summer events/thunderstorms).

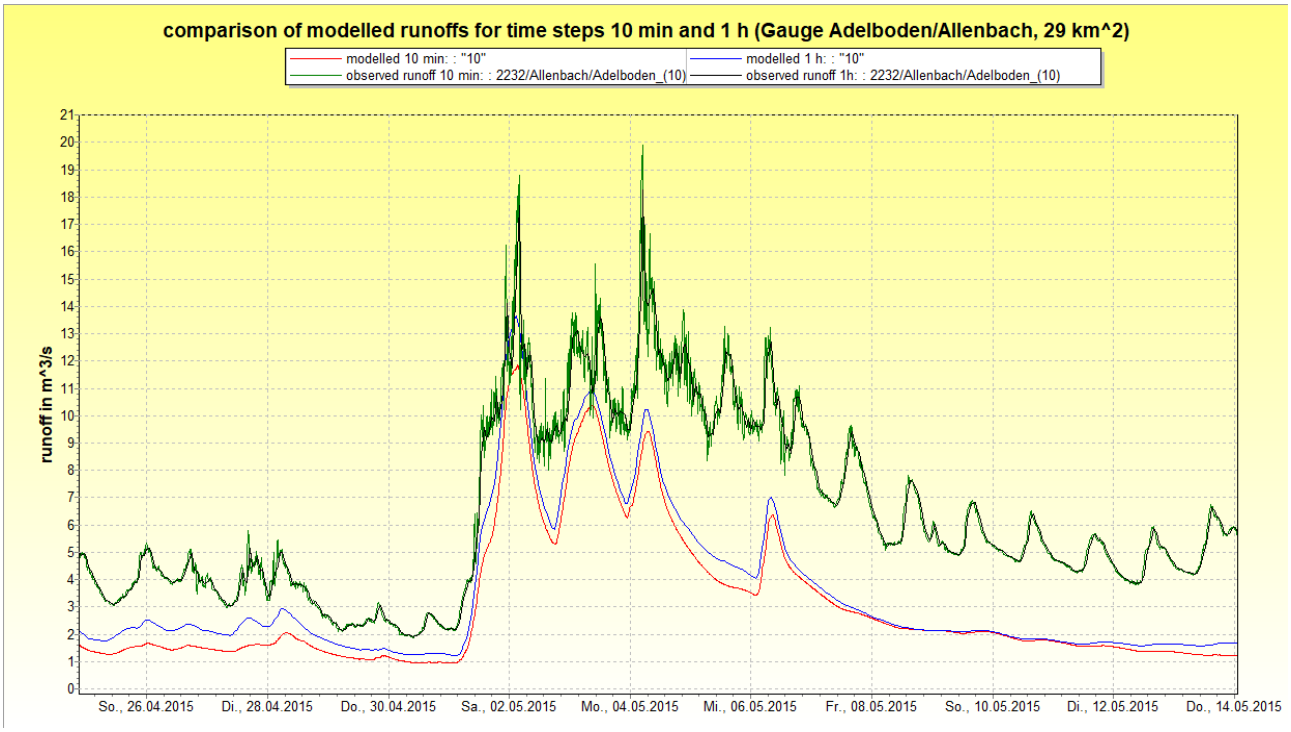


figure 4.14.3.3: comparison of modelled and observed runoff for a 29 km2 subbasin of the river Kander April/May 2015. Time series in green and black are observed runoff data with time step 10 minutes (green) and 1h (black), red (10min) and blue (1h) are modelled runoffs

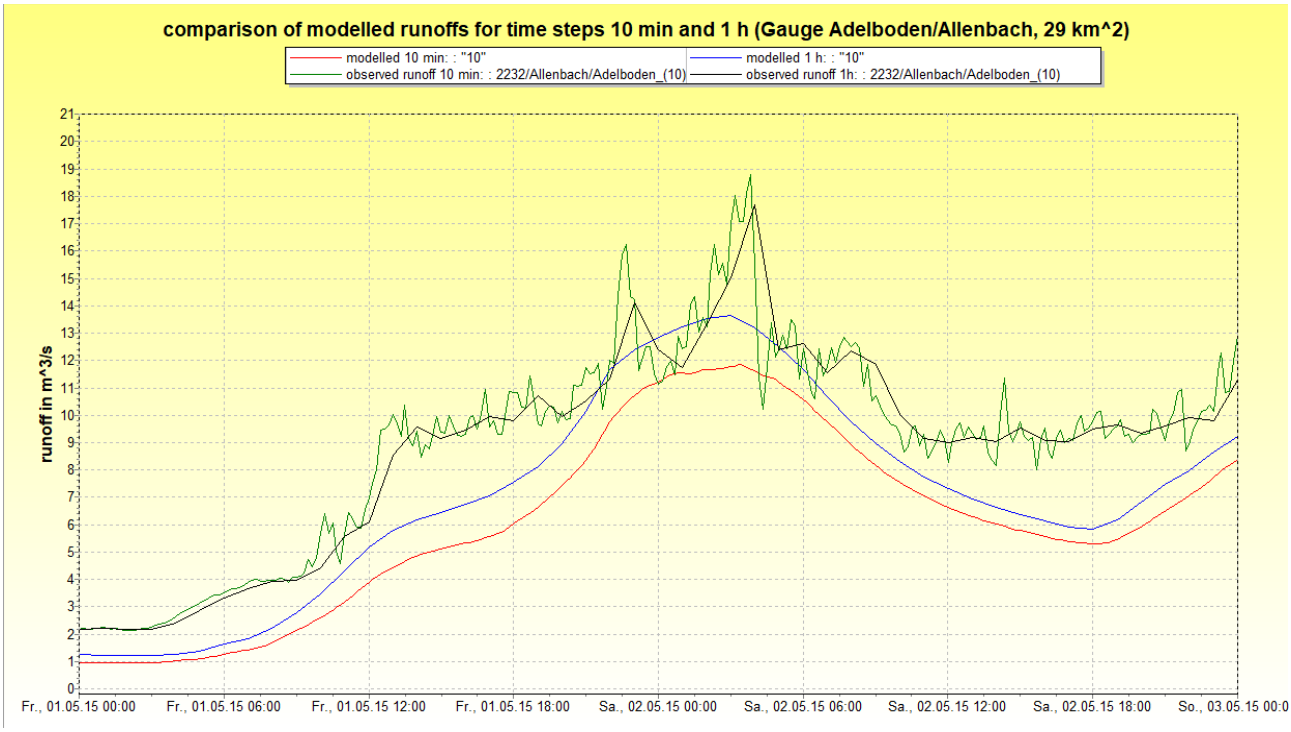


figure 4.14.3.4: comparison of modelled and observed runoff for a 29 km2 subbasin of the river Kander May 1st to May 2nd, 2015. Time series in green and black are observed runoff data with time step 10 minutes (green) and 1h (black), red (10min) and blue (1h) are modelled runoffs

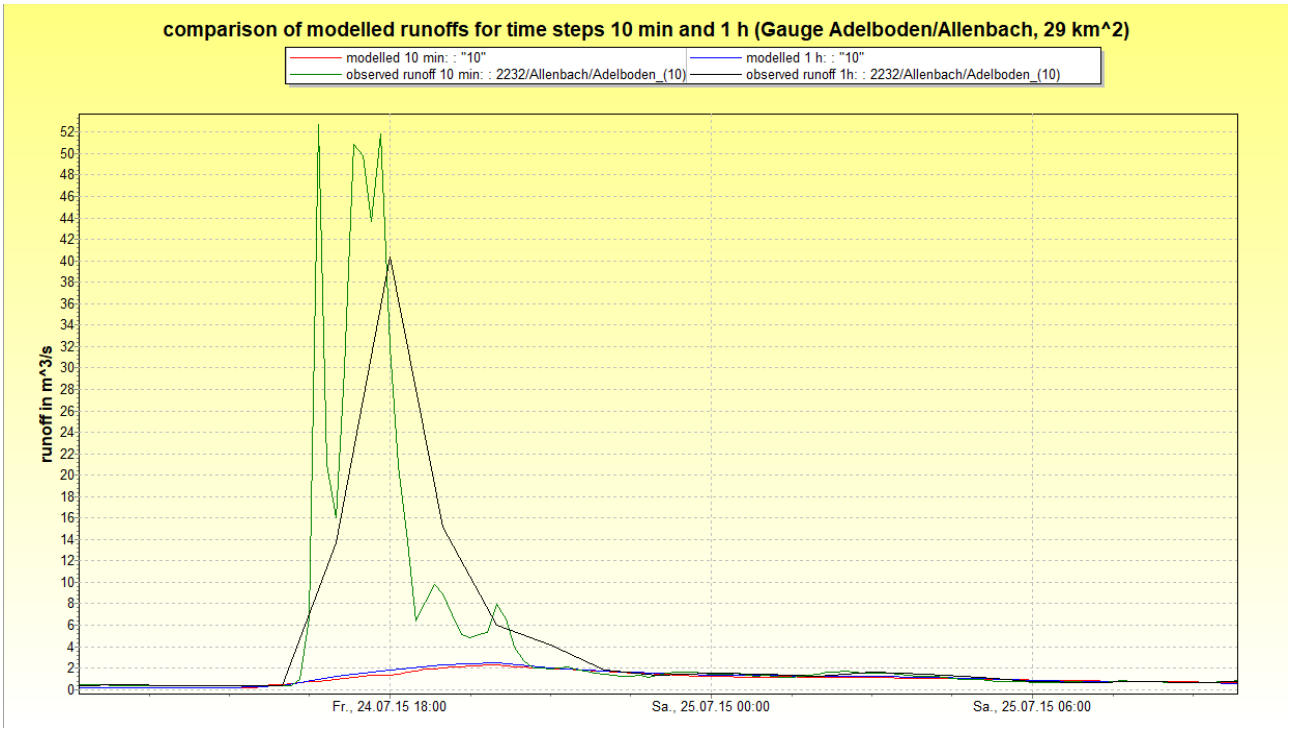


figure 4.14.3.5: comparison of modelled and observed runoff for a 29 km2 subbasin of the river Kander July 2015. Time series in green and black are observed runoff data with time step 10 min (green) and 1h (black), red (10min) and blue (1h) are the modelled runoffs

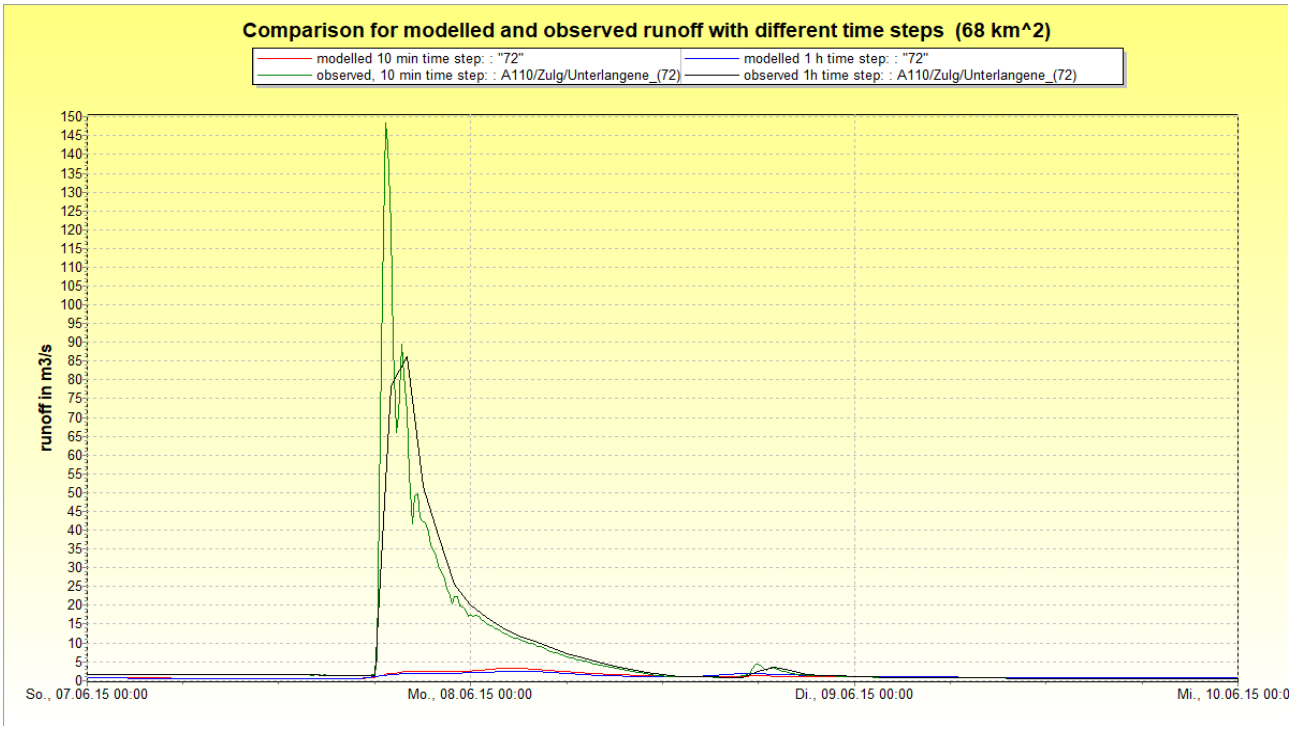


figure 4.14.3.6: comparison of modelled and observed runoff for a 68 km2 subbasin of the river Aare, June 2015. Time series in green and black are observed runoff data with time step 10 min (green) and 1h (black), red (10min) and blue (1h) are the modelled runoffs

4.15 Sensitivity Analysis

4.15.1 Introduction

This section deals with a sensitivity analysis of only the most important model parameters. Only those parameters are considered which have a large possible value range, which are not or only difficulty observable, and which are empirical parameters. Mainly the parameters of the soil model of model version 1 and 2 (1=Topmodel, 2=Richards) will be looked on. The sensitivity of the interpolation algorithms will not be investigated. The parameters for precipitation correction, which have more or less a linear impact on the results are not considered. The radiation correction does not use empirical parameters whereas the temperature modification does not impact the runoffs of larger basins. Therefore both modules are also not examined. Further, the relations between sunshine duration and radiation can be fitted very exactly using measurements, so these parameters are also no subject to a sensitivity analysis. The routing parameters are in principal observable and they are relatively insensitive compared to the soil model parameters. Therefore, those routing parameters are also not considered here.

If possible and reasonable, the sensitivity of the model to the investigated parameters is expressed using the model efficiency criteria. The examples are taken from the Thur basin in Switzerland and from the Wernersbach basin in Germany (Saxonia).

4.15.2 Parameters of the evaporation model

The potential evaporation is usually calculated using the Penman-Monteith -approach (Monteith, 1975). The simplified methods after Hamon (Federer and Lash, 1983) resp. Wendling (1975) are not considered here, also the parameters for modifying temperatures of day and night. The most difficult parameter of the evaporation equations is the surface resistances r_s. Table 4.14.1 shows the differences in the evaporation components when varying the resistances between 25 % and 200 % of the optimum value, which was obtained by calibrating the results on the observed water balance (precipitation P = 1523 mm). The results are taken from model runs for the Rietholzbach basin for 1984.

Table 4.14.1: impact of the minimum stomata resistance r_s on evaporation and discharge

%-value of optimum r_{sc} →	25 %	50 %	75 %	**100 %**	125 %	150 %	175 %	200 %
ETP (pot. evapor. [mm])	688	612	557	**515**	481	453	430	409
ETR (real evapor. [mm])	674	609	556	**515**	481	453	429	409
EI (interception evap. [mm])	143	139	136	**133**	130	128	126	124
Q (runoff [mm])	844	909	961	**1003**	1037	1066	1088	1108
R^2 (runoff observ. vs. model)	0.764	0.784	0.794	**0.795**	0.788	0.777	0.762	0.747

If the resistances are set to 50 %, the real evaporation rises by around 19 % ($ETR_{100\%}$ = 515 mm, $ETR_{50\%}$ = 556 mm), whereas an increase of r_s by 50 % leads to a decrease in evaporation by app. 12 % ($ETR_{150\%}$ = 453 mm). A doubling of r_s leads even to a decrease by 20 % ($ETR_{200\%}$ = 409 mm).

d

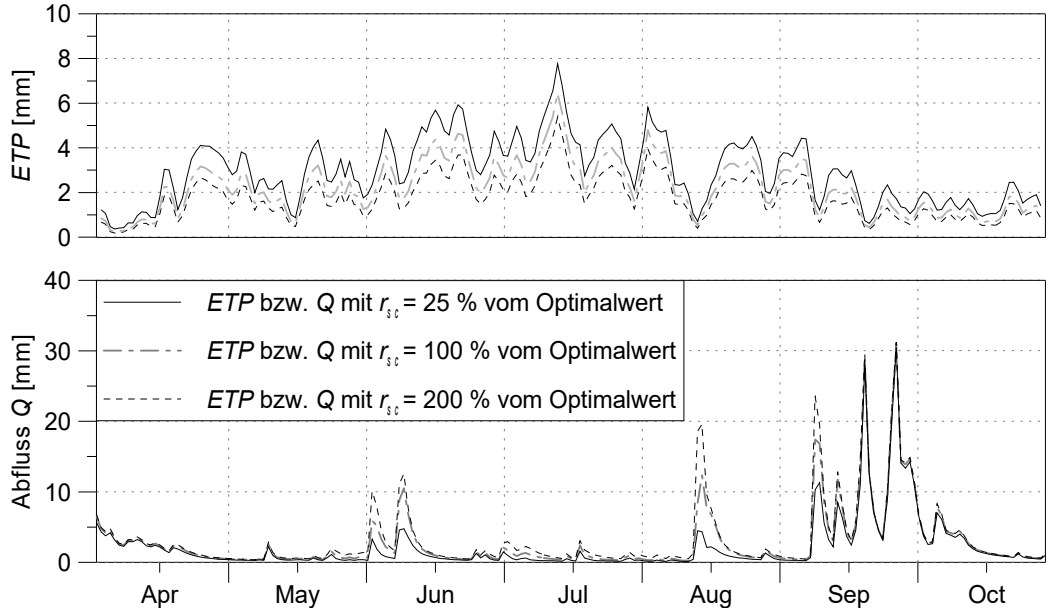


figure 4.15.2.1: model sensitivity on changes in the surface resistances rs, using results from the Rietholzbach basin for 1984; units of ETP and discharge are in mm/day

The distribution of the R^2 –values in table 6 shows similarly good results for resistances with 50 % to 125 % of the "optimum" 100 %-values. Comparing the values of observed and modeled discharge it is obvious that resistances with 75 % of the "optimum" values would give a better water balance, but with a slightly lesser R^2. This example demonstrates the importance of looking also to balances and time series (e.g. modeled versus observed hydrographs) in order to validate the parameters got by a R^2 –optimized calibration.

Figure 4.15.2.1 shows the evaporation and the modeled total discharges as results of 25 %-, 100 %- and 200 %-r_s-values, all graphs printed as 3-day-moving averages. The very small resistances of the 25 %-variant leads to more soil water extraction resulting in much smaller flood events following the precipitation events (June, August, first September event, solid line). Only when the soil is saturated after the first September event, the flood peaks are much more similar. Using higher values for the surface resistances, less water evaporates and precipitation events will generate much higher floods, like the event in mid August.

Figure 4.15.2.1 and table 4.14.1 show, that the surface resistances are very important for the total amount of evaporation. If these parameters can not be calibrated e.g. by comparisons with lysimeter observations or by Bowen-ratio-measurements, they must be calibrated very carefully.

4.15.3 Parameters of the interception model

Interception occurs on leaf surfaces as well as on the vegetation litter layer. Also storage on the soil surface in hollows or puddles should be considered by this approach (in WaSiM-version 2 an additional possibility for considering storage in hollows is given by using a pond-grid which defines a maximum surface water storage capacity for each grid cell). For the interception storage the bucket approach is used. The capacity of this storage depends on the vegetation coverage degree v, on the leaf area index LAI, and on the maximum thickness of the interception water layer h_{SI}. Vegetation coverage degree and leaf area index are in principle observable variables, but h_{SI} is an

"effective" parameter which has to be calibrated because its only an approximation to think of the interception as a storage which is filled and emptied uniformly all over the area wit the same rate.

Table 4.14.2: dependency of evaporation components on the interception storage capacity h_{SI} ($h_{100\,\%}$ = 0.35 mm, data from model runs for the Rietholzbach-basin, year 1984; ETR: real evaporation and transpiration taken out of the soil, EI: interception loss, ETP potential evaporation)

h_{SI} as %-age of h_{opt}		25 %	50 %	75 %	**100 %**	125 %	150 %	175 %	200 %
ETR	[mm]	515.3	515.2	515.1	**515.0**	514.9	514.9	514.9	514.8
ETR - EI	[mm]	468.9	433.3	404.6	**381.7**	362.4	345.4	330.4	318.0
EI	[mm]	45.8	81.5	110.1	**132.9**	152.3	169.2	184.2	196.6
EI/ETR	[%]	9 %	16 %	21 %	**26 %**	30 %	33 %	36 %	38 %

Also the emptying of the interception storage differs between model and reality. The model evaporates with a uniformly potential rate all over the area, but in reality the interception storage consists of separate drops which even may evaporate with higher than the potential rate because of the higher saturation vapor pressure over curved surfaces compared to plain surfaces. However, under humid climates the value of h_{SI} is only of minor importance for the long term water balance but initial losses for flood generation may be considerably even there. Table 7 shows the results for the eveporation components as annual sums for different maximum storage capacities (Rietholzbach-basin, 1984). It is obvious that the different capacities affect only the ratio of interception evaporation (*EI*) and transpiration (*ETR*), the total amount of evaporation remains more ore less unchanged at 515 mm. Using a parameter value of 0.35 ... 0.4 mm for h_{SI}, the interception evaporation gets a realistic fraction of around 25 % ... 30 % on the total evaporation leading to a storage capacity for pine forest of around 4 mm.

4.15.4 Parameters of the snow model

The snow model uses 4 effective parameters if using the temperature index approach:

- the threshold temperature for snow melt $T_{0,m}$
- the temperature $T_{R/S}$, at which 50 % of the precipitation are falling as snow and 50 % as rain
- the temperature range for the transition from snow to rain $2 \cdot T_{trans}$
- the degree-day-factor *TGF*

These parameters partially affect the models reaction in opposite directions or they depend on another, like the threshold-temperature for snow melt $T_{0,m}$ and the degree-day-factor *TGF*. A higher value for the threshold temperature would also result in higher degree-day-factors to get the same snow melt rates.

the temperature range for the transition from snow to rain T_{trans} and the central value of this range, the "equilibrium" temperature $T_{R/S}$:

These parameters are especially important, if the temperature in the entire model domain is within the range of $T_{R/S}$-$T_{trans} \leq T \leq T_{R/S}$-$T_{trans}$. This is unprobable for mountainous regions but it may more frequently happen in low land regions. There, a large value of T_{trans} has the effect, that even if the mean temperature is clearly above or below the melt threshold, there are still solid and liquid fractions at the same time, thus avoiding sharp transitions from snow to rain in space as well as in time. This may help in avoiding model errors due to erroneous temperature measurements or interpolations. However, if the temperatures are everywhere in the model domain above $T_{R/S} + T_{trans}$ or everywhere below $T_{R/S} - T_{trans}$ the parameter T_{trans} is meaningless.

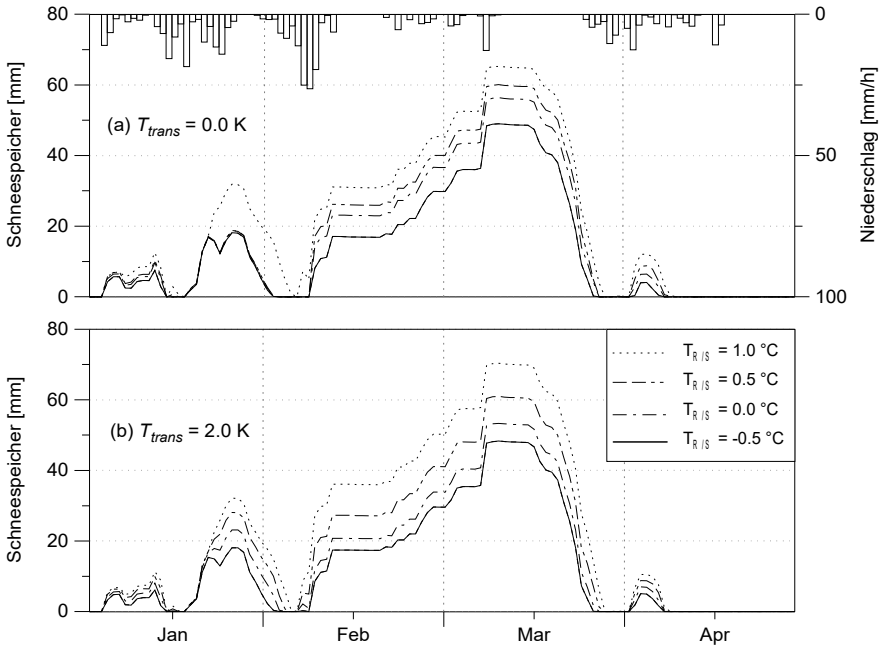


figure 4.15.4.1: Snow accumulation at various threshold temperatures TR/S (snow/rain) (a) Ttrans = 0.0 K; (b) Ttrans = 2.0 K, Rietholzbach catchment, 1984 "Schneespeicher" means snow water equivalent; "Niederschlag" means precipitation

Figure 4.15.4.1 shows a comparison of the temporally course of the snow water equivalent, once modeled using a 2 K transition zone from snow to rain (b) and once without this effect (a). The period from January, 20th shows the impact of the transition zone on the spatial and temporal distribution of snow: In case (a), with the transition zone $T_{trans} = 0$, either it melts and no snow at all is accumulated for temperatures above app. 0.5 +°C ($T_{R/S} < T$), or all precipitation is accumulated as snow for temperatures below 0.5 °C ($T_{R/S} > T$). On the other hand, case (b) shows a smooth transition from accumulating to melting conditions using a transition zone of 2 K. If the temperatures are below the transition zone, like in figure 4.15.4.1 for the period of mid February to mid March, the parameter T_{trans} has no effect. It's also apparent from figure 4.15.4.1 that the value of $T_{R/S}$ affects also the mean snow water equivalent due to different snow fractions. The optimum values for the Thur-Basin are $T_{R/S} = 0.6$ °C and $T_{trans} = \pm 1$ K.

snow melt threshold temperature $T_{0,m}$ and degree day factor TGF:

The melt parameters used in the temperature index approach are mainly the melt threshold temperature $T_{0,m}$ and the degree day factor *TGF*, which must be cross-calibrated because of their impacts on each other. Figures 4.15.4.2 and 4.15.4.3 shows these effects. Figure 4.15.4.2 shows the impact of the degree day factor on the snow melt for a threshold temperature of $T_{0,m} = +0.5$ °C (a) and for a threshold temperature of $T_{0,m} = -1.5$ °C (b), respectively. Particularly at the end of the melting period the differences between small and large *TGF*-values are obvious, resulting in a two week difference of the melt periods end if comparing *TGF* = 1.2 mm·°C^{-1} ·d^{-1} and *TGF* = 2.4 mm·°C^{-1} ·d^{-1} for case (a). On the other hand, a lower threshold value of $T_{0,m}$ –1.5 °C results in a much more independent course of the snow melt.

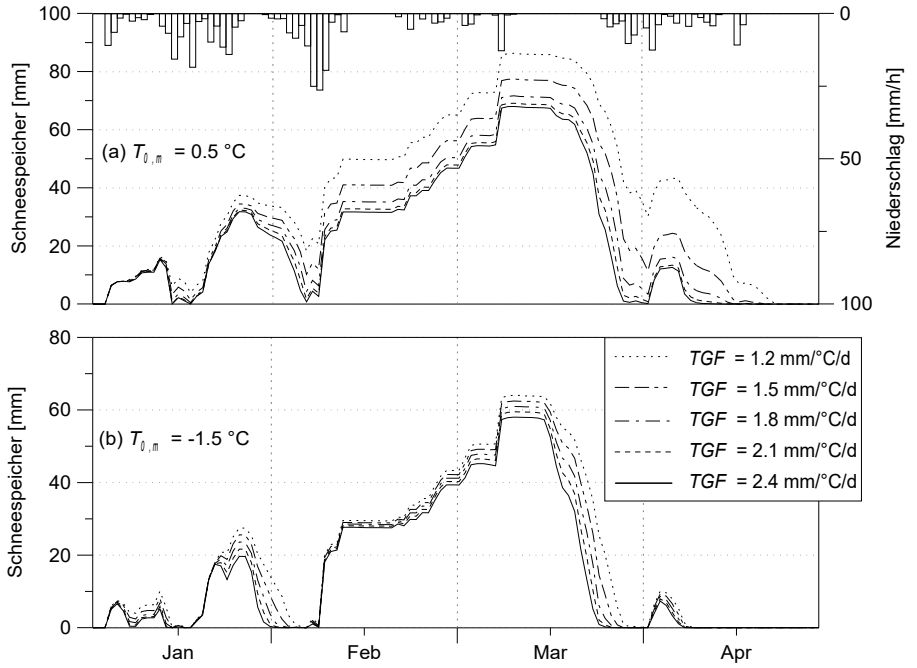


figure 4.15.4.2: temporal course of snow water equivalent for different degree day factors: (a) melt threshold temperature T0,m = +0.5 °C, (b) T0,m = -1.5 °C, Rietholzbach basin, 1984; "Schneespeicher" means snow water equivalent, "Niederschlag" means precipitation

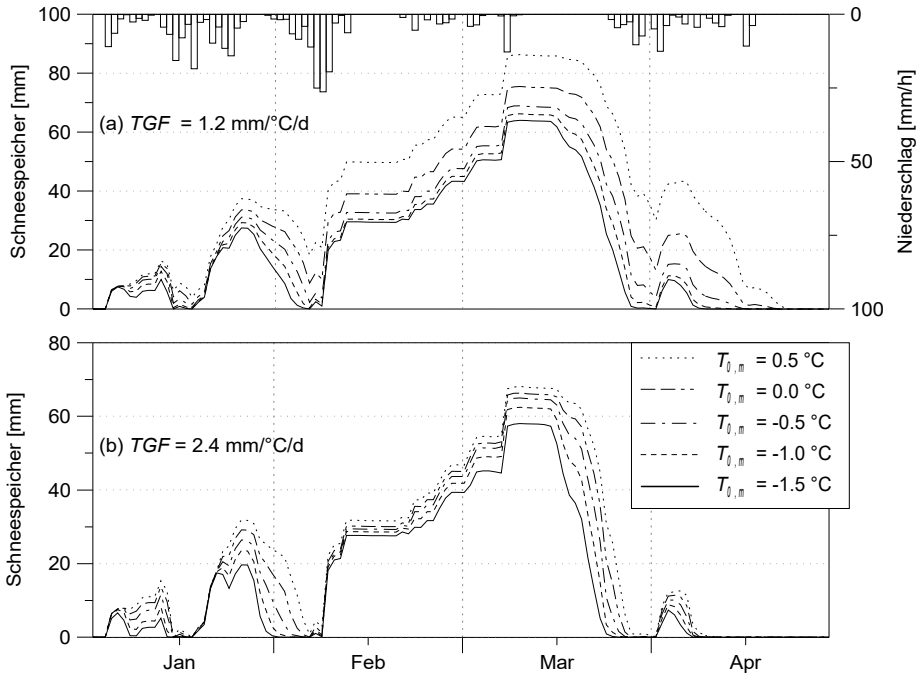


figure 4.15.4.3: temporal course of snow water equivalent for different melt threshold temperatures T 0,m at different degree day factors, Rietholzbach basin, 1984; "Schneespeicher" means snow water equivalent, "Niederschlag" means precipitation

Also figure 4.15.4.3 shows the impact of the melt threshold $T_{0,m}$ on the course of the snow water

equivalent. The optimum parameter values for the Thur basin were found to be TGF = 1.8 mm·°C^{-1} ·d^{-1} and a threshold temperature $T_{0,m}$ = -0.5 °C. Compared to the literate, e.g. Braun (1985) who found 4 to 6 mm·°C^{-1} ·d^{-1} to be optimal, the value of TGF is rather small, but this is also due to the low threshold value as well as due to the temporal resolution of one hour for the calibration runs, so the diurnal course of the temperature is more effective for melt than using a daily temperature average.

4.15.5 Parameters of the soil model for model version 1 (Topmodel-approach)

The soil model of model version 1 (without infiltration submodel) is controlled by 9 parameters and 2 initial conditions:

- m recession parameter for base flow [m]
- T_{korr} correction factor for the transmissivity of the soil [-]
- K_{korr} correction factor for vertical percolation [-]
- k_D single reservoir recession constant for surface runoff [h]
- SH_{max} maximum storage capacity of the interflow storage [mm]
- k_H single reservoir recession constant for interflow [h]
- P_{grenz} precipitation intensity threshold for generating preferential flow into the saturated zone [mm·h^{-1}]
- r_k scaling of the capillary rise/refilling of soil storage from interflow [0…1]
- c_{melt} fraction on snowmelt which is surface runoff [0…1]

The initial conditions are the content of the interflow storage and the initial saturation deficit. The recession constants for surface runoff and interflow may be estimated using observed hydrographs. If no interflow is considered (SH_{max} = 0) this is true also for the TOPMODEL-recession parameter m. But if interflow is used, then m must be calibrated, too, because then m is not derivable from observed hydrographs anymore. The percolation scaling factor K_{korr} makes sense only in areas with deep soils with a deep groundwater table. In such basins, often a first flood peak is followed by a second flood which is usually much slower rising and falling but which may be even higher than the first peak.. Because WaSiM was not applied yet to such basins, a sensitivity analysis using observed runoff data was not processed. In most basins, this effect may be neglected and there K_{korr} should be set to a sufficient large value to avoid water logging or even an hydraulic disconnection to the saturated zone (e.g. by setting K_{korr} to 1000).

The following examples cover the impact of the most important parameters of the soil model: the recession parameter m, the correction factor for the transmissivities T_{korr}, the “refilling” and capillary rise scaling factor r_k, and the threshold precipitation capacity for generating macro pore runoff P_{grenz}. Some of these parameters are not independent from others, so they usually should be cross-calibrated. For example, the recession parameter m and the scaling parameter T_{korr} have strong impacts on each other. Therefore it is important to use not only observed discharges for the calibration but also, if available, internal state variables like groundwater table depth’s and/or soil moistures.

recession parameter m and transmissivity correction T_{korr}

Although the recession parameter m may be estimated by analyzing observed hydrographs (see eq. 66) this parameter is used here for the sensitivity analysis, also because of the estimation of m is sometimes not even possible due to too much impact of frost or evaporation on the discharge or because of interflow is considered. Also it often depends on the modeler how to define recession periods which are the basis for the estimation of m. The more clear and long recession periods, which have to be unaffected by frost, evaporation and snow melt, can be analyzed, the more confidential is the value of m.

This becomes even more complex in basins where the interflow cannot be neglected. Here, also no

interflow should be present during the recession periods for the estimation of *m*. Recessions during the summer are bad candidates because their corruption by evaporation, whereas wintry recession periods often are influenced by snow melt or frost. From fall to spring also the impact of interflow is usually significant or there are only too less or too short recession periods. In the end this leads to a calibration of *m*, using the estimation after equation (66) as a first minimum estimation – usually the use of interflow requires at least a doubling of *m*.

The effect of *m* is as follows: The smaller *m* (and therefore also the tangent of the inverted discharges), the more water leaves the soil, leading to a fast decreasing of the "groundwater table" and thus to a fast increasing of the saturation deficit. But, on the other hand, a larger saturation deficit decreases the outflow from the saturated zone. In conjunction with higher recharge rates in the winter and higher evaporation losses in the summer this leads to a regime with high base flow rates in the winter and low base flow rates in the summer. Or, in other words, there is only a small long term storage effect of the underground and the baseflow has a large dynamic. Larger *m*-values will result in considerably more temporal storage of water in the saturated zone by transferring water from the winter into the summer. Figure 4.15.5.1 shows the effect of a to small parameter *m*: The winter precipitations generate too much discharge in the winter. This water is missing in the summer, when too less base flow is generated. This is also obvious from the saturation deficit which remains large from begin of May. The same effect can be observed if using different T_{korr}-values varied over a wide range.

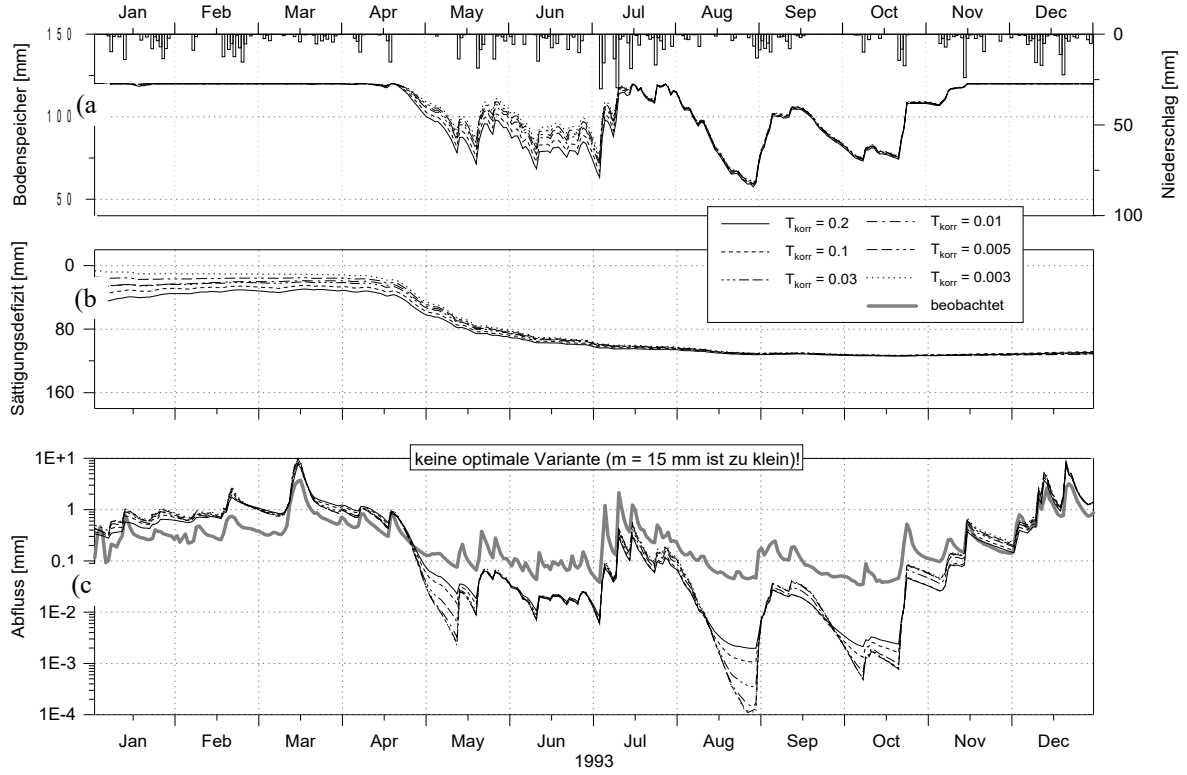


figure 4.15.5.1: model sensitivity on variations in Tkorr if using a too small recession parameter m (15 mm), showed for (a) soil moisture, (b) saturation deficit and (c) discharge; Wernersbach, Saxonia, 4.6 km2, 365 to 465 m a.s.l., hourly values, 1993; Abfluss=discharge, Sättigungsdefizit = saturation deficit, Bodenspeicher=soil moisture, beobachtet=observed

A comparison between figures 4.15.5.1 and 4.15.5.2 shows the impact of a variation of the recession parameter *m*. Figure 4.15.5.2 shows a temporally much more balanced saturation deficit than 4.15.5.1, which is nonetheless more intense affected by the scaling parameter T_{korr}.

Following equation (65) the maximum base flow $Q_0 = T_{korr} \cdot e^{-\gamma}$ depends on the mean topographic index γ. From the definition of the topographic index (eq. 55) follows, that both, the topography as well as the transmissivity of the soil affect the maximum base flow of the soil. Because of the resolution of the digital terrain model has an important impact of the distribution function of the topographic index, usually a shift of this function is required to reach reasonable base flow rates after equation (65). Also the impact of the transmissivity which is usually not exactly measured must be scaled. Both effects can be considered by the scaling factor T_{korr} which affects the base flow generation linearly but the location of the topographic index distribution function logarithmically. Thus, the possible range of values for T_{korr} is between approximately $10^{-5} \leq T_{korr} \leq 10^{+5}$.

The parameters m and T_{korr} depends on another, so they must be calibrated at the same time (so called cross-calibration). Because of the wide range T_{korr} may be within, both parameters should be estimated very carefully.

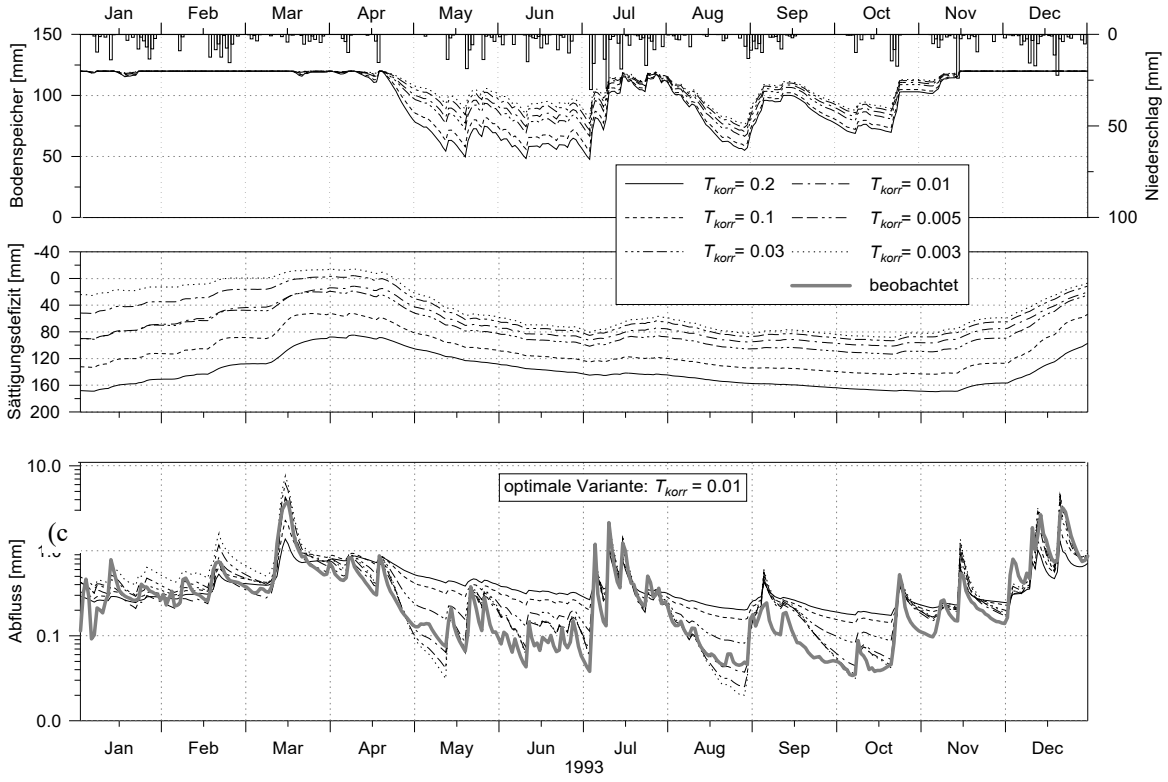


figure 4.15.5.2: model sensitivity on variations of Tkorr at a optimum recession parameter m (55 mm), showed for (a) soil moisture, (b) saturation deficit and (c) discharge; Wernersbach, Saxonia, 4.6 km2, 365 to 465 m a.s.l., hourly values, 1993, Abfluss=discharge, Sättigungsdefizit = saturation deficit, Bodenspeicher=soil moisture, beobachtet=observed

A large transmissivity of the soil leads to a temporally more evenly distributed regime than a small transmissivity because even at low groundwater tables there can flow out considerably amounts of base flow. On the other side, this leads to a high dynamic of the groundwater table (or the saturated zone), resulting also in high recharge values after heavy precipitation events and thus in relatively small floods. This can be shown for the graph of $T_{korr} = 0.2$ in figure 4.15.5.2. Since the depth of groundwater (or the saturation deficit) is an important factor for modeling the capillary rise from the groundwater into the soil storage, both parameters m and T_{korr} affect the average soil moisture and thus the real evaporation in cases of dry periods. This effect can also be observed by comparing figures 4.15.5.1(a) and 4.15.5.2(a)

Using the model efficiencies (linear as well as logarithmic, explained variances as well as R^2) an optimum can be found for both parameters. A series of simulations should be run with various values for m and T_{korr}. A result of such a series is shown in figure 4.15.5.3. The model efficiencies are interpolated by a smooth surface, the optimum parameter set for the Wernersbach-basin could be found at $T_{korr} = 0.018$ and $m = 0.054$ resulting in a model efficiency (linear R^2) of around 0.95.

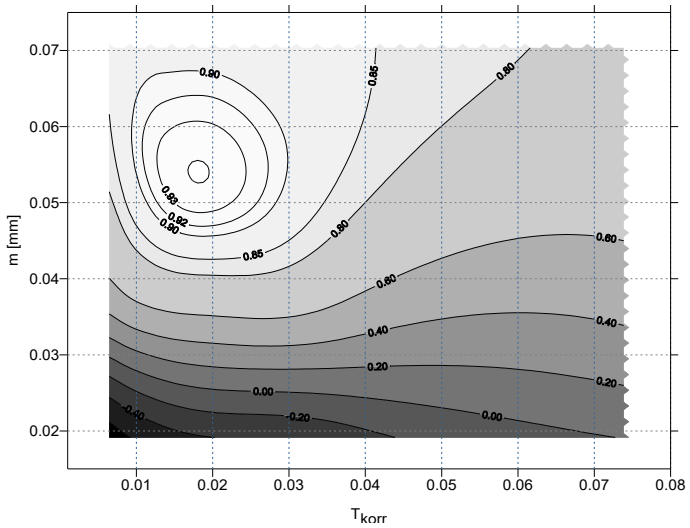


figure 4.15.5.3: distribution of model efficiencies (R2) as function of Tkorr and m; Wernersbach, Saxonia, 4.6 km2, 365 to 465 m a.s.l., daily time step, 1993

precipitation intensity threshold P_{grenz} for generating macro pore percolation and scaling factor for capillary rise r_k

Both parameters, P_{grenz} as well as r_k influence the filling and emptying of the plant available soil water storage, the saturated zone and the interflow storage. P_{grenz} is responsible for the generation of macro pore runoff, which is routed directly into the saturated zone. Even if the soil is only partly saturated, a fraction of the precipitation (dependent on the precipitation intensity) can be used to fill both the saturated zone storage and the interflow storage. Thus it's possible to explain rising interflow and base flow rates even if the soil storage is still not saturated after a heavy rain event. This concept is a extension of the infiltration model as well as of the TOPMODEL, which presume homogenous soils without macro pores. The water infiltrated by the infiltration model is thus only partial stored in the soil storage.

During the summer, often a decrease in base flow can be observed for small watersheds which is stronger than the usual decrease by base flow and interflow recession. This effect can be explained by the additional extraction of water from the unsaturated zone as well as from the interflow storage by plant roots at shallow groundwater tables. WaSiM version 1 models this effect in a conceptual way whereas WaSiM version 2 can consider this effect implicitly by applying the RICHARDS equation. The variable saturation approach of the TOPMODEL version allows a temporally and spatially distributed modeling of shallow groundwater regions which are in contact with the soil water storage. Using the local saturation deficit, the local depth of the saturated zone is compared to the depth of the root zone for each grid cell in each time step. When there is an intensive contact between the soil water storage and the saturated zone, a high rate of capillary rise and reflow from the interflow storage into the soil moisture will be possible. Thus, a moist basin will allow high reflow and capillary rise rates whereas a dry basin will virtually not allow any recharge at all.

Because of the conceptual scheme of the TOPMODEL the rate of capillary rise must be scaled using the parameter r_k which is a calibration parameter. The parameters r_k and P_{grenz} affect the models reaction in opposite directions. So both parameters have to be investigated together. The impact of the "capillary rise" on the total water balance rises with increasing fractions of evaporation on the entire water balance. For regions with small evaporation fractions the soils are usually not sufficiently dry to allow much capillary rise. There, both parameters are relatively unimportant. However, the following example of the Wernersbach shows, that capillary rise may well be important for both, total evaporation and runoff.

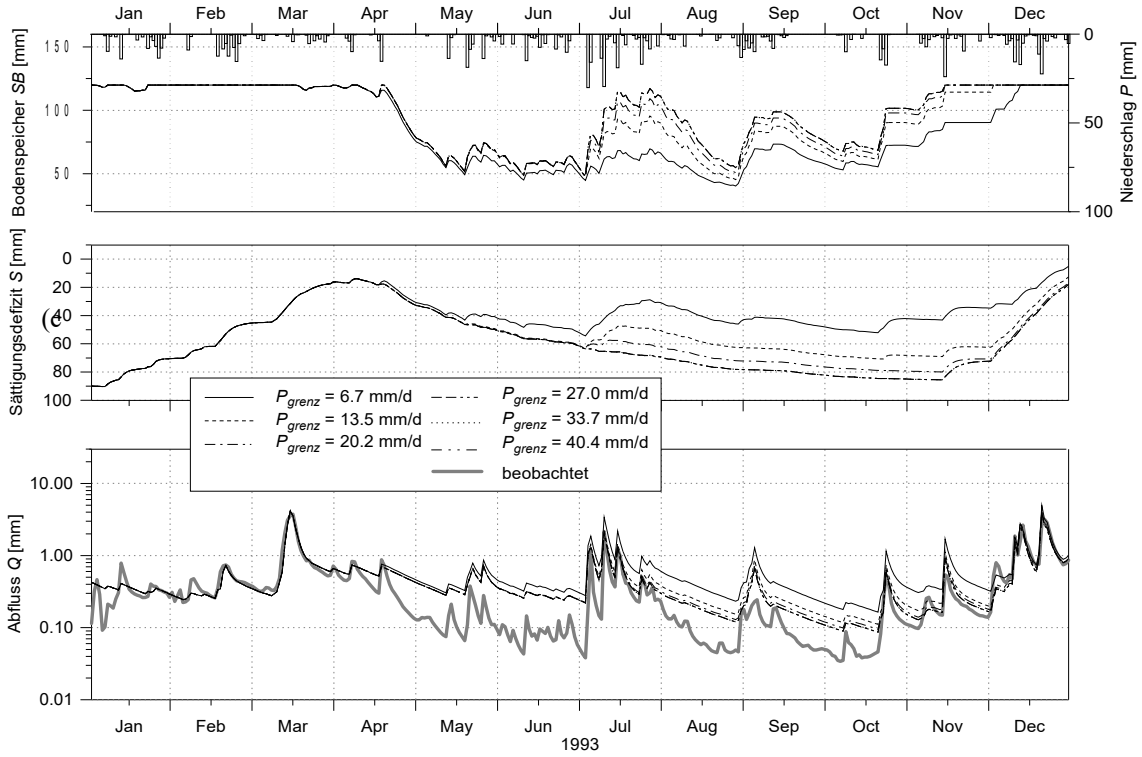


figure 4.15.5.4: Impact of the threshold precipitation intensity Pgrenz for generating macro pore drainage on (a) soil moisture, (b) saturation deficit and (c) runoff; without capillary rise/reflow from interflow into soil storage (rk = 0.0); Wernersbach, Saxonia, 4.6 km2, 365 to 465 m ü.M., daily time steps for 1993; "Abfluss"=discharge, "Sättigungsdefizit"=saturation deficit, "Bodenspeicher"=soil storage, "beobachtet"=observed

Figure 4.15.5.4 shows the impact of the precipitation intensity threshold P_{grenz} on soil moisture, saturation deficit and runoff. All model runs were carried out without capillary rise and reflow from interception storage ($r_k = 0$). The effect are much too high summery discharge rates because of not considering the uptake-processes which are important in this basin. This affects also the soil moisture which decreased dramatically, regardless of the value of P_{grenz}. Figure 4.15.5.4 makes it clear how the parameter P_{grenz} affects the separate storages: The smaller P_{grenz}, the less water can infiltrate in the soil matrix and the less water can be extracted by evaporation. At the same time the saturation deficit is decreasing, the groundwater table rises, leading to higher base flows and also to higher interflow rates. It's also apparent from figure 4.15.5.4 that there will be a characteristic maximum value for P_{grenz} valid for a given precipitation regime, because larger values of P_{grenz} will not change the model reaction anymore. In the example of figure 4.15.5.4 this is a value of 27 mm·d^{-1}. Higher precipitation intensities were not found for 1993 in this watershed.

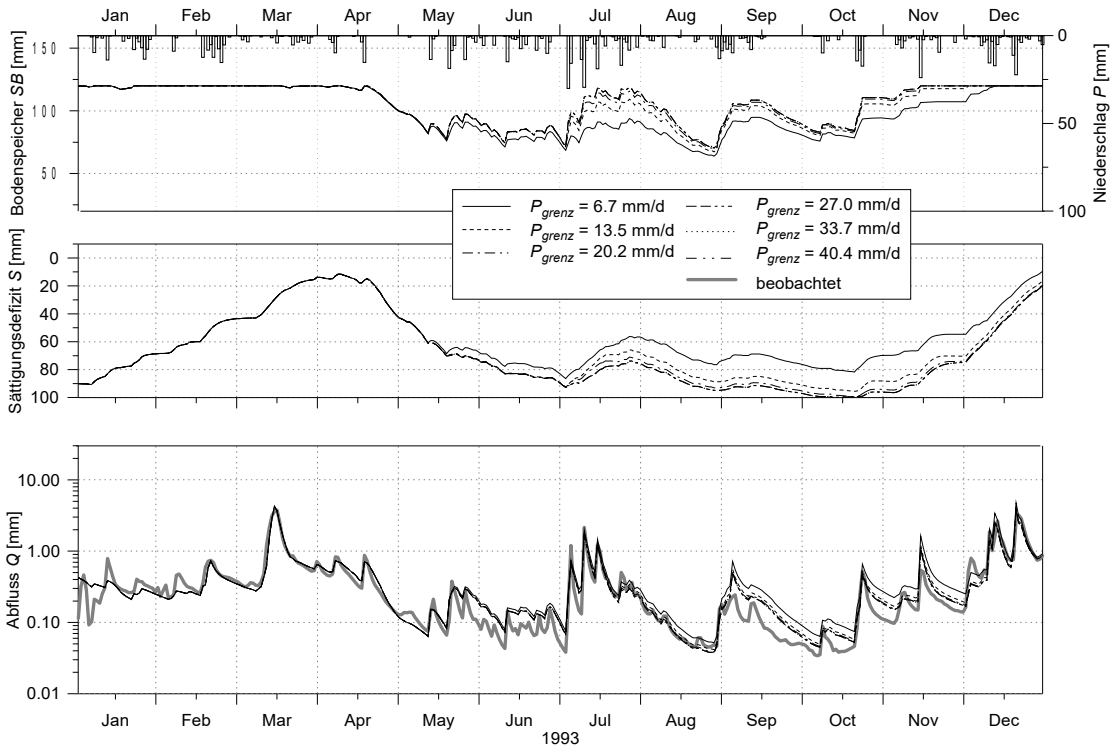


figure 4.15.5.5: Impact of the threshold precipitation intensity Pgrenz for generating macro pore drainage on (a) soil moisture, (b) saturation deficit and (c) runoff; without capillary rise/reflow from interflow into soil storage (rk = 0.0); Wernersbach, Saxonia, 4.6 km2, 365 to 465 m ü.M., daily time steps for 1993; "Abfluss"=discharge, "Sättigungsdefizit"=saturation deficit, "Bodenspeicher"=soil storage, "beobachtet"=observed

Comparing the courses of soil moisture, saturation deficit, and discharge in figures 4.15.5.4 and 4.15.5.5 it becomes clear how important the processes of capillary rise resp. reflow from saturated zone and interflow storage into the soil storage may be. Compared to the graphs without these effects, the soil moisture in figure 4.15.5.5 is much higher leading to much less days with reduced evaporation conditions. At the same time the saturation deficit is larger leading to less base flow.

The parameter P_{grenz} has similar impacts in figures 4.15.5.4 and 4.15.5.5 Soil water storage and groundwater table can be affected but the absolute effect is damped considerably by the capillary rise effect. This means that high intensity precipitation which flows directly into the groundwater may return into the soil storage due to the risen groundwater table and thus due to higher capillary rise rates. In the above example r_k is set to 1.0, the maximum value. The reflow from the interflow storage is calculated after the capillary rise is considered if there are still unsatisfied amounts of eveporation losses which have to been filled up. The potential reflow from the interflow storage is scaled down according to the relative content of the interflow storage itself and according to the parameter r_k.. The scheme of calculating capillary rise and interflow storage reflow is explained by equations (63) and (64). The vertical component, the capillary rise, is only considered if there is a n hydraulic connection between the saturated zone and the soil water storage. This criterion is checked by comparing the actual saturation deficit against the threshold saturation deficit, which is determined by the product of the scaling parameter r_k, the drainable porosity n_e and the root depth z_w.

4.15.6 Parameters of the soil model for model version 2 (Richards-equation)

Because of its more physical base there are much less sensitive calibration parameters for model version 2 using the RICHARDS-equation than for model version 1 using the TOPMODEL-approach. The soil properties, in particular the porosities, the k-Θ- and the ψ-Θ-relations should be understand as fix parameters which should not be calibrated at all (may be they could be replaced by measurement results). Finally there are only 3 parameters which may be calibrated:

- the recession constant k_{rec} for the saturated hydraulic conductivity K_s with depth after equation (2.14.7)
- the interflow drainage density d_r, see equation (2.14.11)
- mainly for larger time steps and larger layer thickness': the maximum relation of the relative hydraulic conductivities of neighboring layers c_k after equation (2.14.9)
- if the groundwater model is disabled, the base flow parameters Q_0 and $k_{ELS,QB}$ must be calibrated; this is not considered here because the application of the groundwater module makes this obsolete

The recession constant k_{rec} considers the decreasing hydraulic conductivity with increasing depth. The conductivities are reduced down to 2 m depth, below 2 m the value remains constant and is taken from the 2 m level (2 m are a fix depth fixed in the model internally). If the soil parameterization is not done using the VAN GENUCHTEN parameters n, α, K_s, Θ_r, and Θ_{sat} look-up-tables for the saturated conductivities of all layers as well as look-up-tables for the k-Θ- and ψ-Θ-relations must be applied. Then. the parameter k_{rec} is not used, but the conductivity profile must be known.

The interflow drainage density d_r considers the "river density" in a grid cell as well as the difference between horizontal and vertical hydraulic conductivity. It is an effective parameter which linearly affects the amount of interflow (see equation (2.14.11)). A model presumption is, that for each grid cell where interflow can be generated (slope > 0) there is at least one drainage channel. This limits a physically realistic model application to grid cell sizes larger than the natural or artificial drainage density. But since this is a rather theoretical limit, the model can also be applied to smaller cell sizes, considering interflow but no lateral interconnections via water flow between neighboring grid cells.

The maximum relation of the relative hydraulic conductivities of neighboring layers c_k was introduced into the model in order to compensate or even avoid numerical problems which arise from the vertical and temporal discretization of the RICHARDS-equation. Unless in other models, WaSiM uses constant time steps of e.g. 1 hour or 1 day and relatively thick layers of 0.1 to 0.5 m, so the discretization may be to coarse. The parameter c_k limits the effective k_S-value used for the flux calculation between two layers up to a minimum value in order to avoid in dry soils a permanent water logging. Without this parameter this could be the result, if a heavy rain would fall onto a dry soil. The dry second layer would have a much too small relative conductivity, so the water would fill up the first layer and then surface runoff would be generated. Since this effect may also occur in the reality, the parameter c_k is a calibration parameter which should be chosen according to the region, the precipitation pattern and other environmental conditions. It's value depends on the vertical discretization of the soil and on the temporal discretization. It ranges between <50 (Thur-basin, time step 5 days, 33.3 cm soil layers) up to >100 (Thur-basin, time step 1 hour, 33.3 cm) but may be even smaller or larger for other regions of the world.

Chapter 3.7.3 also shows an example of the model sensitivity to this parameter.

4.15.7 Spatial resolution

The spatial resolution to use for a given project depends mainly on the spatial structure of the model

domain. A measure of the required spatial resolution may be the characteristic morphologic length, also called the macroscopic length or the typical slope length. The altitude-dependence of most of the input variables (like temperature, vapor pressure, wind speed etc.) leads also to a resolution which is adequate to consider these dependencies.

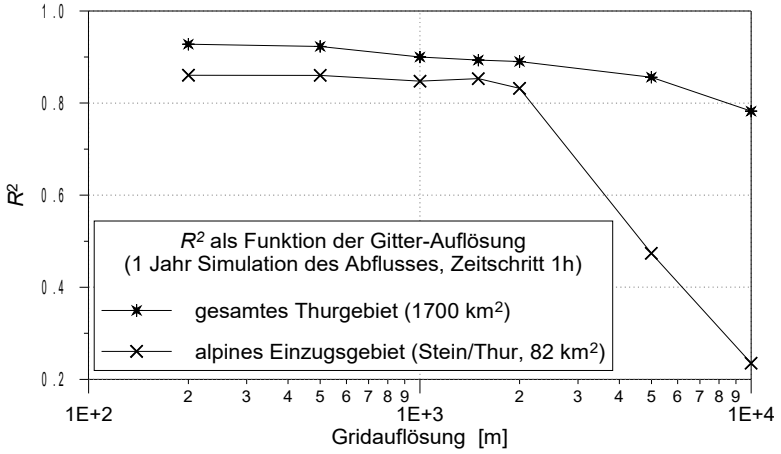


figure 4.15.7.1: Impact of the spatial resolution on the model efficiency, resolution varied between 200 m and 10'000 m, 1 year simulation of discharge, time step 1 hour, the upper graph for the entire Thur basin, the lower graph for the more alpine upper Thur basin at Stein/Thur.

The most efficient resolution (balance between model efficiency (accuracy) and computer run time) may be found by a series of model runs using different spatial resolutions. Figure 4.15.7.1 shows the effects of different spatial resolutions on the model efficiencies R^2. It clearly suggests, that the maximum grid cell size in alpine regions should be below 2000 m (better: below 1000 m), whereas the resolutions for flatter areas may be up to 2000 m or 5000 m. Often, a to coarse resolution results in difficulties in parameterization of the model, so it may be better to use e.g. a 500 m resolution instead of 1000 m in order to consider the variations in surface or soil properties.

4.15.8 Conclusion

The following parameters are the most sensitive parameters with respect to the modeled discharges:

- the surface resistances r_{sc} for the vegetation if they are much higher or much lesser than the optimum value
- the threshold temperature rain/snow $T_{R/S}$
- the degree day factor *TGF*
- the threshold temperature for snow melt $T_{0,m}$

soil model of Version 1 (TOPMODEL -approach)

- the recession parameter *m*
- the correction factor for transmissivities T_{korr}
- the scaling factor for capillary rise resp. for reflow from the interflow storage into the soil storage r_k

soil model of model version 2 (RICHARDS -equation)

- scaling of interflow by a drainage density d_r
- maximum allowed relation of relative conductivities for neighboring layers c_k
- recession constant k_{rec} for the saturated hydraulic conductivities (recession with depth)

Not too important for the discharge, but important for partitioning the precipitation water into water

balance components, the following parameters are:

- the maximum capacity of the interception storage h_{SI} (model version 1)
- the temperature transition range from snow to rain T_{trans}
- the threshold precipitation intensity for generating macro pore percolation P_{grenz} (model version 1)

The spatial resolution is important especially for mountainous basins. With increasing cell sizes there is a certain basin dependent threshold the results will become increasingly inaccurate. These thresholds are at around 1000 to 2000 m for alpine catchments and at around 2000 to 5000 m for hilly regions.

4.16 Calibrating the model

4.16.1 Introduction

The process of the model calibration is not a trivial problem which requires hydrologic expertise as well as a sufficient knowledge of the model structure and the model reactions. The following descriptions may be used as a "calibration guide" for inexperienced WaSiM users. Based on the parameter sensitivity analysis of chapter 3.6 a ranking for calibrating parameters is suggested including practical hints in doing the calibration.

If WaSiM is calibrated for a new basin it is generally recommended to use the parameter values of already calibrated basins for the first calibration step, especially for those parameters which are insensitive. Only those parameters which are sensitive or which depends strongly on basin properties should be calibrated then. As showed in the previous sub-chapters, the most important parameters are the soil model parameters for model version 1 (TOPMODEL-approach). In model version 2 (RICHARDS-approach) also the soil model parameters are sensitive. Therefore, the calibration is shown in two examples for the soil model parameters, once for model version 1 and once for model version 2.

All the other parameters of other modules are less important and they can be taken without large errors from other model applications in similar areas (like evaporation or snow melt parameters). Only the routing parameters must be newly estimated for each model application. This is usually be done during the preprocessing using the topographic analyzing tool TANALYS.

4.16.2 Calibrating the soil model for model version 1 (Topmodel-approach)

The soil model of model version 1 is controlled 9 parameters which have to be specified in the control file for each subbasin. Not all of these parameters are equally important, the most sensitive parameters are the recession parameter m and the scaling factor for the topographic indices T_{korr}. These two parameters will now be used for a first calibration step.

first step of the calibration (soil model, model version 1 - TOPMODEL)

- ***parameter identification by hydrograph analysis***

Using observed hydrographs, the estimation of recession constants for surface runoff and interflow is relatively easy. If the original TOPMODEL-approach is used (WaSiM without interflow → $SH_{max} = 0$), then this is also true for the recession parameter m. The estimation of the single reservoir recession constants for k_D and k_H using linear regression fits in logarithmic transformations is a well known practice, so here the focus is laid on the estimation of m.

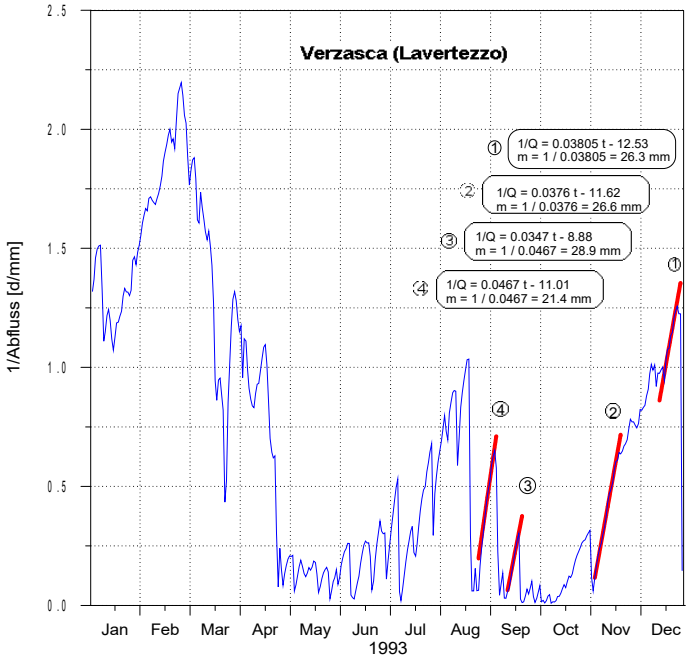


figure 4.16.2.1: Estimation of the recession parameter m of the river Verzasca (Switzerland) for 1993

Parameter *m* can be estimated as the tangent of the slope of invertedly plotted discharge recession periods. These recession periods should not be affected by too much evaporation (because the recessions in hot seasons are much faster due to the evaporation from shallow groundwater areas near rivers). Also impacts by snow melt have to be avoided as well as periods with frozen soils. It is recommended to look for as many of useable recession periods as possible in order to get a parameter value which is as confident as possible. The time step of the observed hydrograph does not significantly influence the result, thus discharges in mm/h or mm/day are equally usable. The examples are shown for the alpine Verzasca-catchment (Ticino, Switzerland). Figure 4.16.2.1 shows the inverted hydrograph for a one-year-period (daily time step). The average value of the tangents of all linear fits is the parameter *m* (for use in WaSiM: convert this result which is given in mm into meter).

Table 4.15.1: recession constant m [m] for the river Verzasca (Switzerland) ;
m = 0.027 m (average value for 13 recessions between years 1993 and 1996)

1993	1994	1995	1996
0.0266	0.0357	0.0277	0.0287
0.0214	0.0281	0.0217	0.0278
0.0263	0.0235	0.0267	0.0256
0.0289			

- ***estimation of T_{korr}***

If the recession constant *m* is estimated as a first approximation (later, *m* must be varied, again), the scaling factor for the topographic indices has to be determined. There are two possibilities:

a) parameter estimation using specific basin information's

b) parameter estimation by optimizing ("trial and error")

parameter estimation using specific basin information's

If T_{korr} should be estimated in this way, some basin data are required. These data are the mean saturation deficit *S* for the subbasin and the matching observed base flow. To get *S*, which is not

measurable, the mean groundwater table GW and the mean drainable porosity n_e have to be known in order to calculate S by $S = GW \cdot n_e$. In addition, the parameter m (which should now be known as first approximation) and the average topographic index γ should be known. γ can be determined by:

$$\gamma = \frac{1}{n} \sum_{i=1}^{n} \ln \frac{\alpha}{T_0 \tan \beta} \qquad \textbf{(4.15.1)}$$

with γ average topographic index [-]
n number of grid cells for the subbasin [-]
a local specific catchment area (a=A/cell size) [m]
T_0 hydraulic transmissivity [m²/s]
β local slope [°, rad]

Base flow is calculated in the model by:

$$Q_B = Q_0 \cdot e^{-S/m} \qquad \textbf{(4.15.2)}$$

with Q_B base flow [mm/time step]
Q_0 maximum base flow at full soil saturation [mm/time step]
S average saturation deficit of the subbasin [m]
m recession constant [m]

with Q_0 expressed by:

$$Q_0 = e^{-\gamma_{korr}} \qquad \textbf{(4.15.3)}$$

and $$\gamma_{korr} = \ln\left(f \cdot T_{korr}\right) - \gamma \qquad \textbf{(4.15.4)}$$

with γ_{korr} corrected average topographic index
f unit scale factor for consideration of the unit of discharge and for conversion of the transmissivity units from m²/s into mm²/time step.
if time step = 1 hour: $f = 3.6 \cdot 10^9$, if time step = 1day: $f = 8.64 \cdot 10^{10}$

Equations (4.15.4) and (4.15.3) can be integrated into equation (4.15.2) . This results in the equation for determining the scaling factor T_{korr}:

$$T_{korr} = \frac{Q_B}{f \cdot e^{-\gamma - S/m}} \qquad \textbf{(4.15.5)}$$

example: Verzasca (Switzerland)

A baseflow of Q_B = 1.3 m³/s was observed for a 186 km² area (this is an equivalent of 0.6 mm/day). The average topographic index is γ = 21.97, the recession parameter was estimated to m = 0.027 (see table 4.15.1). The average saturation deficit was estimated to be S = 0.06 m, as given by an estimated mean groundwater table of 1.5 m below the surface and a drainable porosity of 0.04. The scaling parameter T_{korr} is then:

$$T_{korr} = 0.6 / 8.64 \cdot 10^{10} / e^{-21.97} / e^{-0.06/0.027} = 0.22 \qquad \textbf{(4.15.6)}$$

The value of T_{korr} depends approximately linear on the average groundwater table below the surface. After the first estimation of T_{korr} by equation (4.15.5) it usually must be varied and optimized. However, virtually always the required data for determining T_{korr} by specific basin properties and states are not available. In such cases, the data may be estimated or T_{korr} must be determined by an trial and error approach.

<u>*parameter optimizing*</u>

The optimization is done in a first step by varying T_{korr} in a series of runoff simulations. All other parameters are constant for all optimization runs. To give an orientation table 4.15.2 shows preliminary parameter values, except for m and T_{korr}.

Table 4.15.4: possible initial parameter values, example for a Vercasca basin (Switzerland)

parameter	value	comment
m [m]	(0.027)	estimation by hydrograph analysis, see fig. 18; usually also m must be varied, especially if the recessions are not unique or if interflow is considered
K_{korr} [-]	1000	The correction factor scaling the drainage from the unsaturated into the saturated zone should be sufficiently large in order to avoid the breakdown of the hydraulic interconnection between unsaturated and saturated zone. If K_{korr} would be too small and T_{korr} too large, the saturation deficit becomes larger and larger whereas the percolation from the unsaturated zone becomes very, very small due to the exponential recession of the soil conductivities with increasing saturation deficit. In such cases the water will fill up the soil an the unsaturated zone and all rain will become surface runoff. Another reason that K_{korr} should be rather large is the fact that percolation uses also macro pores and other preferential flow paths, but the hydraulic conductivity of the soil is given for vertical matrix flow only.
K_D [h]	5	from hydrograph analysis or taken from similar catchments
K_H [h]	50	from hydrograph analysis or taken from similar catchments, at $SH = 0$ this parameter does not make any effects, because no interflow is modeled
SH_{max} [mm]	0	interflow is not used here, so it must be set to 0; later this will change
P_{grenz} [mm · h^{-1}]	10	large precipitation threshold value, in order to avoid any fast macro pore drainage (all precipitation infiltrates into the root zone soil storage)
r_k [0…1]	0	no consideration of capillary rise and reflow from the interflow storage into the soil storage
c_{melt} [0…1]	0	no consideration of surface runoff from snow melt
SUZ_0 [mm]	0	initial content of the unsaturated zone storage (large pores above field capacity)
SD_0 [n · nFK]	1.5	initial saturation deficit: affects the initial base flow; with 0 as initial value very much base flow is generated until the model is in equilibrium. Its better to use a small value then a large value because too large values may cause the model to simulate a hydraulic disconnection (like described for the K_{korr} -parameter). If this happens, SD_0 must be reduced.

The possible range of the scaling factor T_{korr} is between approximately 10^{-5} and 10^{+5}. Because the parameter impacts the distribution function of the topographic indices logarithmic, the shift of the distribution function along the x-axis may be in the range of app. –9 to +9. Experiences in using WaSiM show a range of $10^{-3} \leq T_{korr} \leq 10^{+3}$ (logarithmic between –6.9 … +6.9). If no a-priori information about T_{korr} are available (e.g. from estimations using average groundwater table, see above), the range of variation of T_{korr} has to be very large. The model efficiencies (R^2, EV) are used to estimate the parameter quality. Because the number of required simulation runs may be high it may be better to use daily time steps instead of hourly time steps. Figure 4.16.2.2 shows the results for the optimizing process for various T_{korr}-values. Obviously, an optimum for T_{korr} should be looked for between 0.1 and 1. In a second optimizing step, the search can be refined to get a better estimate for the optimum value, see figure 4.16.2.3.

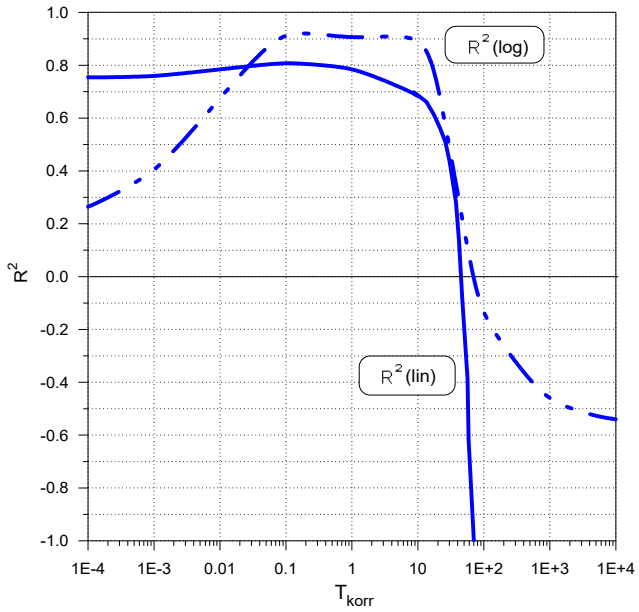


figure 4.16.2.2: 1st optimizing step: distribution of the model efficiency R2 (linear and logarithmic) as reaction on variations in Tkorr; Verzasca, Simulation in daily time steps for 1993

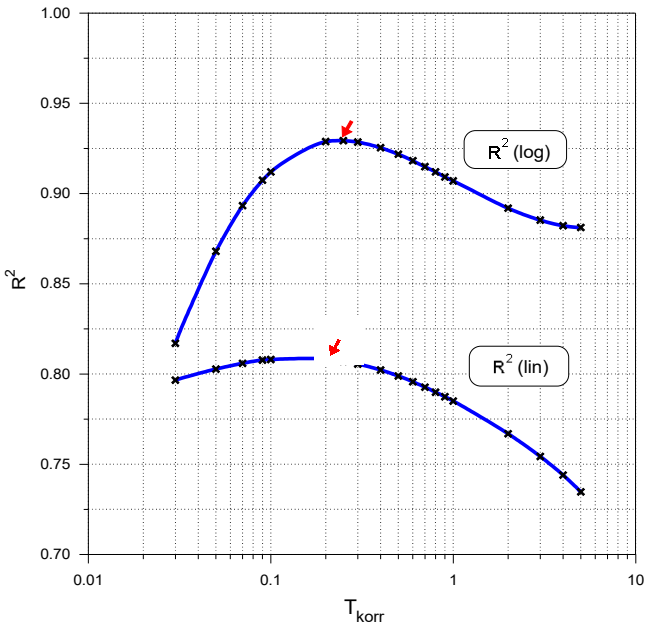


figure 4.16.2.3: 2nd optimizing step: distribution of the model efficiency R2 (linear and logarithmic) as reaction on variations in Tkorr; optimum for Tkorr at around 0.25 for R2(log); Verzasca, Simulation in daily time steps for 1993

Figure 4.16.2.4 shows the effects of the parameter T_{korr} on the modeled discharge and on internal states. This underlines the importance of a serious T_{korr}-estimation.

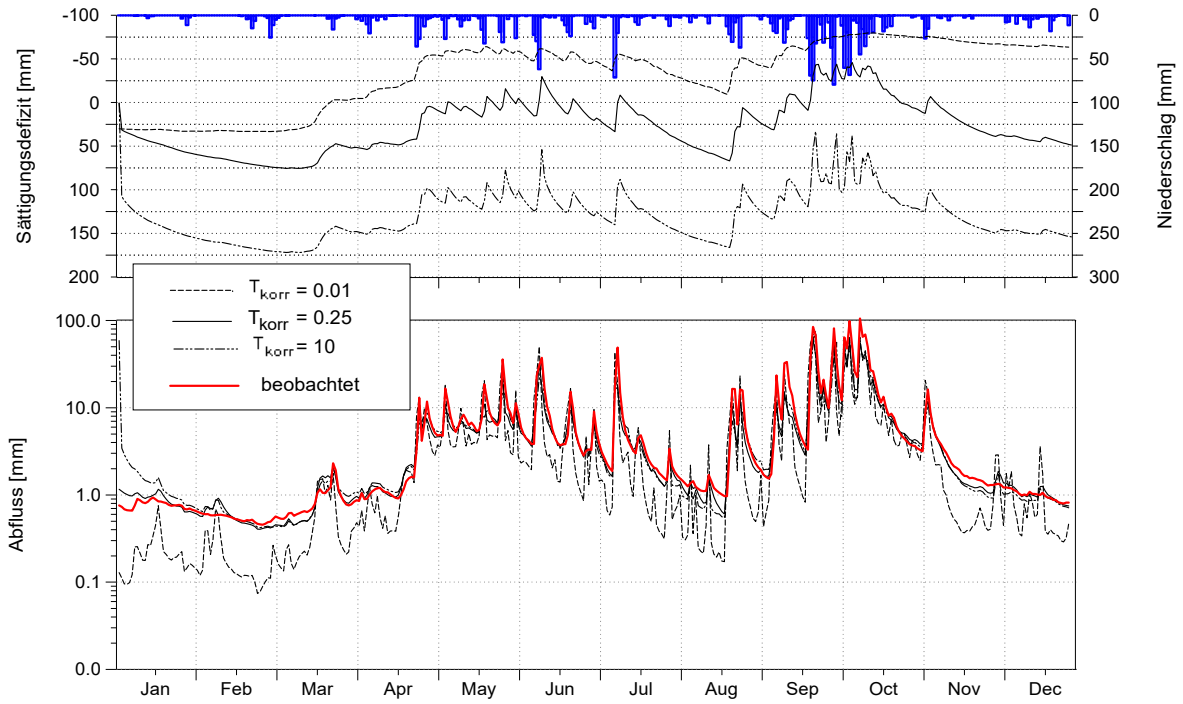


figure 4.16.2.4: model sensitivity on variations in Tkorr; Verzasca, daily time step for 1993; "Abfluss"=discharge, "Sättigungsdefizit"=saturation deficit,"Niederschlag"=precipitation

If a first optimum value for T_{korr} is found and validated by analyzing the runoff components and other internal state variables, the second calibration step may follow.

Note: In opposition to the recession constant *m*, which should be determined for as many subbasins as possible by analyzing observed hydrographs, the first-approximation-T_{korr} can also be used for other, similar subbasins (if they are similar in geology, topography, soils and climate). On the other hand, a further calibration like in the second calibration step should also be done for those subbasins.

Note: In order to avoid impacts of (badly) modeled discharges of first order subbasins on lower subbasins due to the routing, the routing should be done using the option for considering observed discharges. Therefore the first parameter in the [routing_model]-section of the WaSiM control file, the so called "ignorance parameter" must be set to a value of 2, see also chapters 1.10 and 3.5.1.

second step of calibration (soil model, model version 1 - TOPMODEL)

- ***adding interflow***

If the first calibration step does not gave satisfactory results or if the modeler knows, that there must be an interflow component, the interflow should be activated by setting the parameter SH_{max} to values greater than zero. The value of SH_{max} could either be estimated by analyzing observed hydrographs or by taking values from similar subbasins or even by estimating a value considering only the personal experience of the modeler. Current model applications show, that an interflow storage of 10 to 40 mm may be appropriate, so an initial value could be SH_{max} = 20 mm.

Using interflow means that there are now three runoff components instead of only two. The parameter *m* affects now only the base flow, which now is much less dynamic than the mixed interflow/baseflow controlled by *m* was before. Because *m* and T_{korr} depend now on another, a re-calibration is required. Again, this is done by a series of simulations using various values for both, *m* and T_{korr}. The results are evaluated using the model efficiencies. An optimum can be found usually for different parameter sets as compared to the parameter set without interflow. A result of

such an optimization is shown in figure 4.16.2.5 (see also figure 4.15.5.3). However, it is highly recommended to visualize as much of the results as possible, in particular the modeled runoff components but also modeled soil moisture, evaporation, snow cover a.s.o. for both, the time series of subbasin averages and the temporally averaged or aggregated result grids.

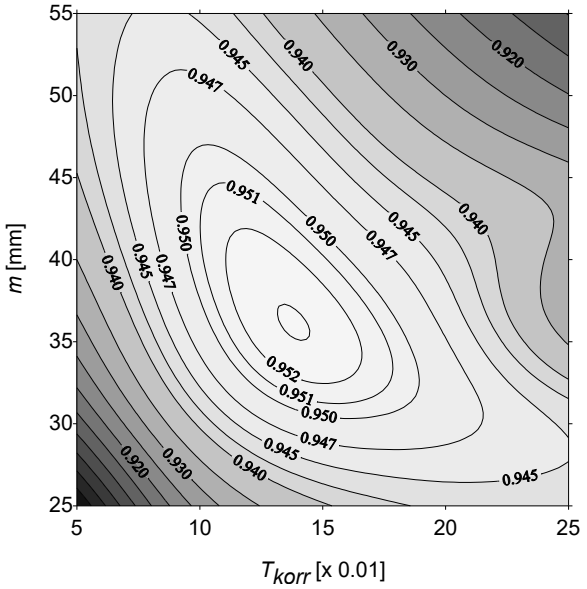


figure 4.16.2.5: distribution of model efficiencies (R2) following variations in m and Tkorr; optimum values at app. Tkorr = 0.14, m = 0.036 m; Verzasca, 1993

If the fraction of interflow on the modeled total flow is unsatisfactory, then variations of the maximum interflow capacity SH_{max} will help to balance the components. A greater SH_{max} usually will lead to increased interflow, whereas a smaller SH_{max} will result in less interflow. However, any changes made to SH_{max} will result in the necessity of re-calibrating the recession parameter m and the scaling parameter T_{korr}. If SH_{max} is changed only in small steps, then this can be reached easily. On the other hand, if peak flow discharges or recession periods are modeled incorrect, then an increase or decrease of the responsible recession constants k_D and k_H may increase the model efficiency.

If required – and usually it is required – this calibration step should include the optimizing of the initial saturation deficit SD_0. This is relatively easy, since each change can immediately be evaluated after only some model time steps by looking at the modeled runoff components and comparing it to the observed runoff. Ideally, the model period starts at a time when the observed runoff consists only of slow subsurface runoff (baseflow), since then the observed runoff can be compared directly with then modeled baseflow which is controlled by the initial saturation deficit SD_0. A well fitted SD_0 increases the total model efficiency R^2 considerably. Another possibility to find a reasonable initial saturation deficit would be a sufficient long time offset from the beginning of the model run until the efficiency criterions are started to estimate. Then the first period is used for a initial stabilization of the model. This can also be reached by running the model twice, the first time with given initial conditions (no grids are read) and written output grids and the second time with read input grids, which are the output grids of the first run.

- ***calibration of the remaining parameters of the soil model***

After calibrating the most sensitive model parameters m and T_{korr}, the remaining parameters should be calibrated. The parameters r_k and P_{grenz} should be calibrated together, because they have partly

contrary effects on the model reaction, see chapter 3.6.5. These parameters are most sensitive, if the model is applied to a rather dry catchment with much higher evaporation than runoff. Thus if working on such basins, the calibration of r_k and P_{grenz} may be especially important. The last parameter to be calibrated is the surface runoff fraction on snow melt c_{melt}. It affects in the first way the modeled discharge in winter and spring, especially the peaks of melt floods. A large c_{melt} value means high surface runoff fractions on snow melt and thus large peak flows in the melt season, but it means also a smaller storage effect of melt water in the soil. Thus, c_{melt} may also be useful to influence the annual storage behavior. The optimum value for cmelt must be between 0 and 1, often it can be set to 0.1...0.2 in flat regions or to 0.2...0.35 in mountainous regions.

4.16.3 Calibrating the soil model for model version 2 (Richards-equation)

Compared to WaSiM version 1, WaSiM version 2 uses a much more physically based soil model. The total number of free parameters is not smaller, but most parameters like the hydraulic properties of the soil (e.g. van Genuchten parameters for k- Θ- and ψ - Θ-relations) are principally observable or derivable from soil type maps. These parameters will not be calibrated here. The calibration parameters, which are empirical parameters, are:

- the scaling parameter for interflow d_r (drainage density)
- the recession constant k_{rec} for the saturated hydraulic conductivity with increasing soil depth
- the maximum relation of relative conductivities between neighboring soil layers c_k

The parameter c_k is relatively insensitive and so it was set to 100 for all calibration runs because it depends mainly on the model resolution in space (soil layer thickness) and time. In addition, the recession parameters for surface runoff and interflow k_D and k_H may be calibrated, but both parameters are relatively insensitive if using a flow time travel grid which makes these parameters even scale independent.

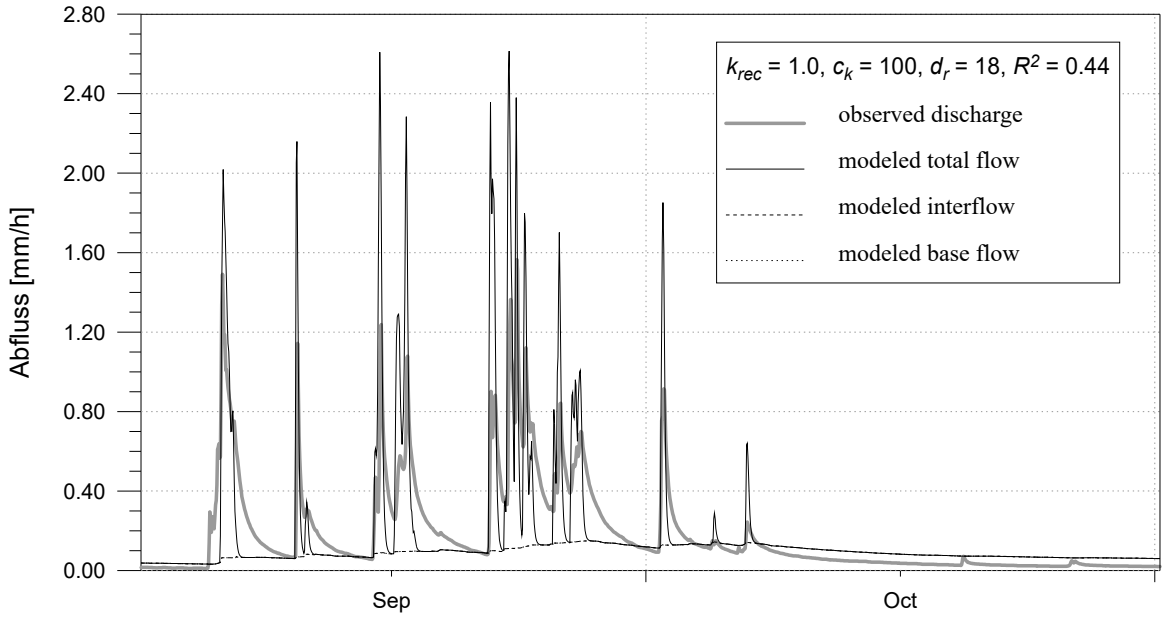


figure 4.16.3.1: Modeled runoff using model version 2 for the Rietholzbach catchment, without interflow

The following example shows the calibration of the soil model for the 3.3 km^2 Rietholzbach catchment, a subbasin of the river Thur. All examples were modeled in a spatial resolution of 500 x 500 m and in a temporal resolution of one hour, so the catchment is represented by 14 cells à

0.25 km^2. The calibration was done modeling the years 1983 and 1984, but evaluating only 1984, so the initial conditions were taken from the last time step of the first model year. The model efficiency criterions are valid for the entire year 1984, not only for the plotted parts.

The first step was a model run without interflow. This can be achieved either by setting $d_r = 0$ or by setting $k_{rec} = 1.0$. The first parameter d_r is multiplied with the potential interflow which depends on the local slope, the water content of the soil and the conductivity. If $k_{rec} = 0$, then there is no conductivity decrease with depth and no lateral runoff components can be generated. The result of this first approximation is shown in figure 4.16.3.1. The dynamic of the runoff is not met, the base flow is increasing too much and the flood peaks are modeled to large. The next step includes the modeling of interflow, which is usually the main runoff component in most northern pre alpine catchments. The interflow scaling parameter is set to $d_r = 18$ as taken from similar catchments. The k_S-recession constant is set to 0.9. All other parameters remain unchanged. The result is shown in figure 4.16.3.2. Compared to fig. 24 the dynamics of the modeled runoff was improved considerably, as can be seen by the R^2 which has increased from 0.44 to 0.8.

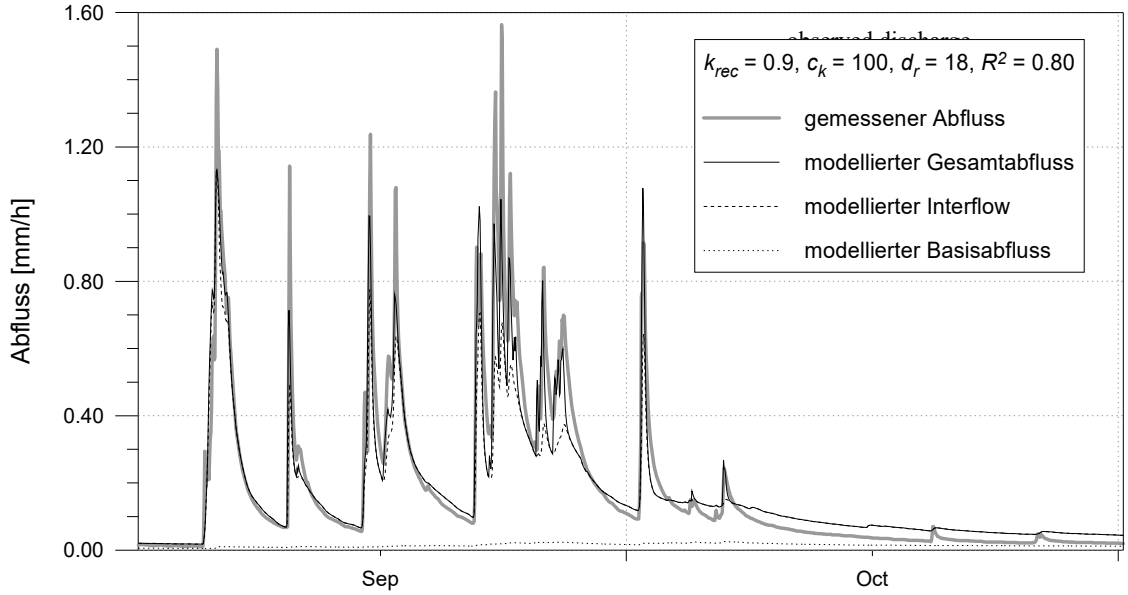


figure 4.16.3.2: Modeled runoff using model version 2 for the Rietholzbach catchment, with interflow, krec = 0.9

The flood peaks, which were too high in figure 4.16.3.2, are now even too small. The interflow is the major runoff component, whereas the base flow virtually disappeared. The interflow (between base flow and total flow) is too high in October. In order to reduce this flow, now the parameter k_{rec} was decreased down to 0.1. Figures 4.16.3.3 and 4.16.3.4 show the model results with values of $k_{rec} = 0.3$ and $k_{rec} = 0.1$, respectively. All other parameters remained constant.

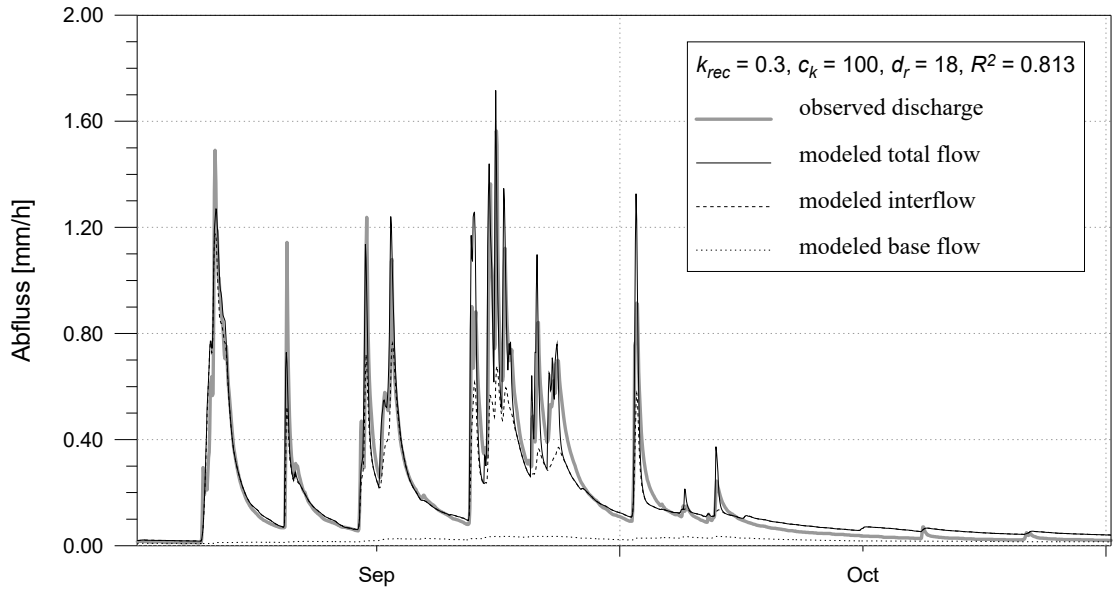


figure 4.16.3.3: Modeled runoff using model version 2 for the Rietholzbach catchment, with interflow, krec = 0.3

As can be seen in the figures 4.16.3.2 to 4.16.3.6 the overall shape of the modeled runoff hydrograph does not significantly change. However, it is remarkable that the flood peaks show a tendency to higher values with decreasing k_{rec}-values. It is also obvious that the recession period in October was modeled much more realistic with smaller k_{rec}-values. But the interflow fraction on the total flow is relatively small in late September, leaving the surface runoff a fraction of partly more than 50 %.

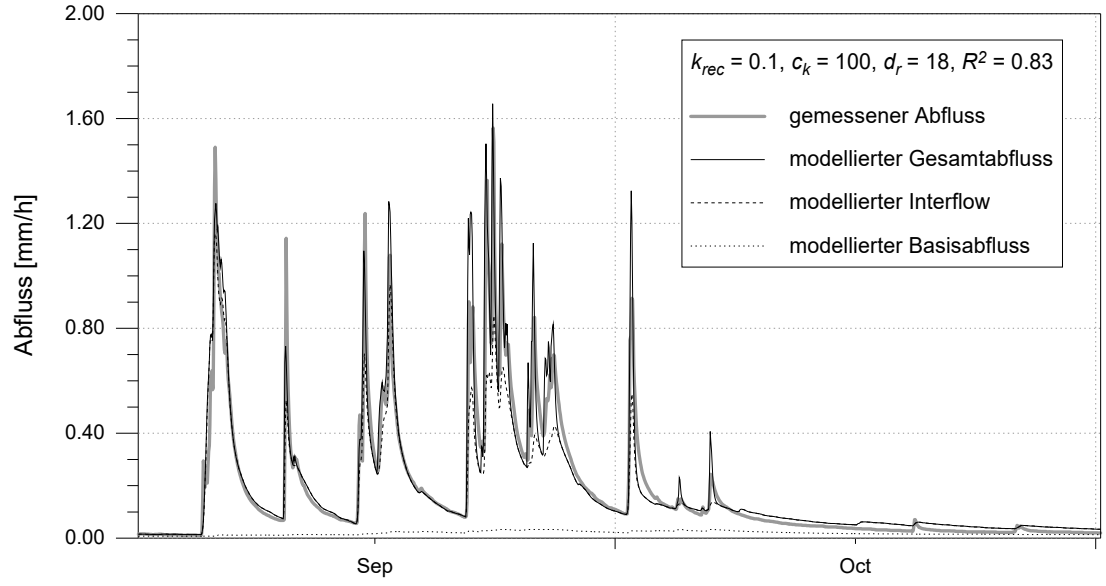


figure 4.16.3.4: Modeled runoff using model version 2 for the Rietholzbach catchment, with interflow, krec = 0.1

Consequently, the next calibration step was the calibration of the interflow scaling parameter d_r. The initial value of 18 was varied between 12 and 30. Figures 4.16.3.5 and 4.16.3.6 show the results with both extremes of the investigated d_r-range.

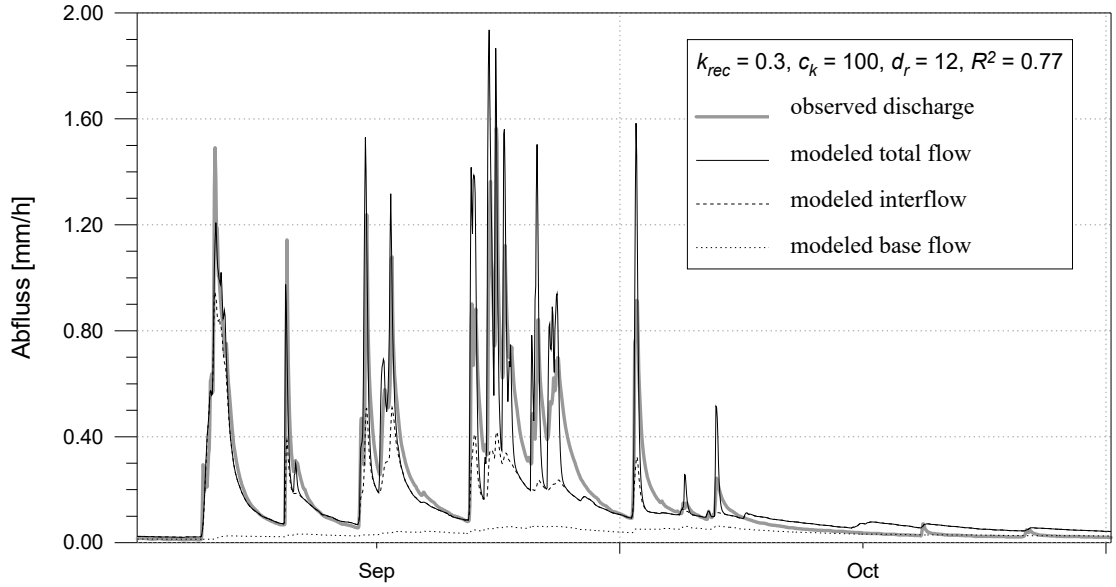


figure 4.16.3.5: Modeled runoff using model version 2 for the Rietholzbach catchment, krec = 0.3, dr = 12

Figure 4.16.3.5 shows a decreasing model efficiency ($R^2 = 0.77$) compared to the former runs due to decreasing d_r from 18 to 12. Not only that the decreased interflow fraction leads to more surface runoff and higher flood peaks, also the base flow in October is increasing again because more water could percolate down to the groundwater. Only an increased drainage density factor d_r leads to the wanted effect of higher interflow fractions on the total runoff and at the same time to smaller runoff values during the recession period. The quality of this parameter set is also underlined by the till now best R^2-values of $R^2_{log} = 0.85$ and $R^2_{log} = 0.89$. The achieved quality of fit is sufficient, so the parameter set can be seen as optimized. However, further small improvements may be possible, but they will be minor compared to the large effect of the above calibrated parameters d_r and k_{rec}. The recession constant $k_{rec} = 0.1$ could also be validated for local soils, which show a recession of the saturated hydraulic conductivity by two orders of magnitude down to two meters depth.

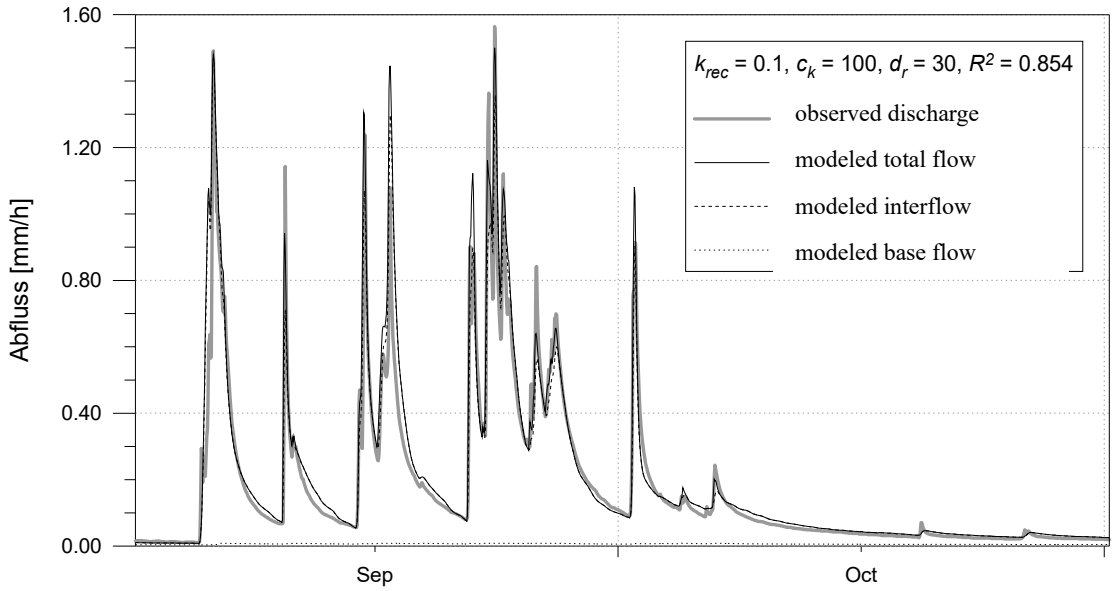


figure 4.16.3.6: Modeled runoff using model version 2 for the Rietholzbach catchment, krec = 0.1, dr = 30

5 Software Tools for Pre- and Postprocessing

5.1 Working with spatially distributed data (grids)

5.1.1 2DSPLINE

Resampling continuous data

command: <2dspline <ingrid> <outgrid> <xll|-1> <yll|-1> <nrows|-1> <ncols|-1> [-f:a|d|s]

This tool uses the bicubic spline technique to resample continuous data. A new cell size can be definied interactively. New base coordinates are possible as well.

As for almost all grid tools, the output format can be chosen by the optional parameter -f:t|s|d, where t stands for text (ASCII), d for double precision binary and s for single precision binary format. The program takes any of these as input format (will be recognized automatically).

See also: RESAMPLE, REFINE

5.1.2 ASCIGRID

Converting ASCII grids into binary grids

command: **ascigrid <ascii-grid> <binary-grid>**

During the conversion a statistic of all valid cell values is calculated and stored in the records 7 to 12 of the binary grid. The statistic parameters are count, minimum, maximum, sum, average, and standard deviation of all valid cells. Nodata values does not count for the statistics. **Note:** The maximum size of a grid depends on the memory of the computer as well as on the data itself. Because the statistics needs an internal sum of all squared values, an arithmetic overflow may happen if the values are very large. This should happen in extremely rare cases only, because the statistic registers are able to deal with numbers up to ±1.0E+4300 (Intel PC).

see also: GRIDASCI, GRIDSURF, SURFGRID, MF2WASIM, GRIDCOLM

5.1.3 ASCII2STACK

Converts a stack from ASCII format into binary format.

Command: Ascii2Stack <input stack in ascii format> <output stack in binary format> [-f:s|d]

The stack ASCII format looks similar to the grid ASCII format:

```
stack_ncols  300
nrows        280
xll          679900
yll          132200
cellsize     500
nodata_value -9999
cnt          0
min          0
max          0
sum          0
average      0
numlevels    15
<values of first cell in first row for all levels>
<values of second cell in first row for all levels>
...cellwise from top left to bottom right
<values of forelast column in last row for all levels>
<values of last column in last row for all levels>
```

WaSiM writes stacks in this format by default (from 9.06.02). To convert this stack into binary format, one of the options -f:s for single or -f:d for double precision should be used (default is double).

5.1.4 FIELDGEN

Generating (log-)normal distributed correlated random fields in ASCII-Grid-file format.

Command: **fieldgen <outfile> <ncols> <nrows> <xll> <yll> <cellsize> <nodata-value> <average> <std.dev.> <corr-length-x> <corr-length-y> [init_value=<init_value>] [<cm_log=0|1>]**

The ASCII formatted output grid is specified by its name <outfile>, its dimensions <ncols> and <nrows> (number of columns and rows), the coordinates of the lower left corner <xll> and <yll>, the <cellsize> and the nodata-value. The grid values are generated randomly according to the given parameters <average> and <standard deviation> of the probability distribution. A correlation length relative to the cellsize have to be specified for each direction. The range for the correlation length is within >0 to 1. In order to get well defined reproducible results, the random number generator can be initialized by the option init_value=<init_value>, with init_value as any number. Each run which uses the same init_value will deliver identical grids. The results are usually normally distributed. However, the option cm_log=1 forces the program to generate a log-normal distribution (LN3). The generated random values are then interpreted to be exponents for the parameters <average> and <standard deviation> to the base 10, so the result grid will contain the powers of the random numbers to the base 10. The lower limit is 0 (third parameter of a 3 parameter log-normal distribution). The statistic shown with GRIDSTAT will therefore not match the given parameters for <average> and <std.dev.>, because it does not refer to the exponents but to the plain values. FIELDGEN can be used e.g. for generating K_f-values with a spatial correlation and anisotropy, e.g. for the WaSiM groundwater module. If the correlation length' are very small, the generated field is spatially virtually uncorrelated and normal (or log-normal) distributed. However, because of the used generation methodology it is better to use the program GRIDGAUS for generating uncorrelated fields, because correlation lengths of 0 are not supported in FIELDGEN. An equal distribution may be achieved by a very large standard deviation, but, again, this should be done with another program, this time the program GRIDRAND, which generates uncorrelated, equally distributed random numbers (e.g. for generating land use distributions). The Program FGM offers all of the above described functionality with a Windows GUI.

see also: GRIDRAND, GRIDGAUS, GRIDSTAT, MAKEGRID, ASCIGRID, FGM

5.1.5 FOCLMEAN

Low band filtering of grid values using a defined window-size in terms of grid cells

Command: **foclmean <ingrid> <outgrid> <window-size>**

For each grid cell a replacing value is generated as average value of all cells within the window centered at the actual cell. The window is moved row by row down the grid. The window size is an odd integer number in terms of grid cells, the smallest window size of 3 (a 3x3 window).

5.1.6 GRIDADD

Adds two grids to a resulting grid on a cell-by-cell basis

Command: **gridadd <ingrid1> <ingrid2> <outgrid> [-f:t|s|d]**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

see also: GRIDMULT, GRIDQUOT, GRIDGRID, GRIDDIFF

5.1.7 GRIDASCI

Converts binary grids into ASCII-formatted grids. The default precision is the optimum precision chosen by the system. However, its possible to specify a precision parameter as last parameter of the program. It may range between 0 and 9 with 9 indicating the optimum precision like the program would generate without precision option. **Note:** A precision of 0 may lead to large errors duo to rounding!

The output is formatted as shown in the example in chapter 3.2.1. The grid values are written row by row downwards like in a table. Separators are tab stops between grid cells and a line feed at the row end. If such files should be transferred between different systems (UNIX ↔ DOS/WINDOWS) the transfer should be done using e.g. ftp or programs which convert line feeds correctly (e.g. unix2dos, dos2unix).

Command: **gridasci <binary-grid> <ASCII-grid> [<significant digits>] [-f:t|s|d]**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

see also: ASCIGRID, GRIDSURF, SURFGRID, MF2WASIM

5.1.8 GRIDCOLM

Converts a binary grid into a x-y-z formatted ASCII file. The ASCII-file contains three columns for x- and y-coordinates and for the z-values, respectively. Each row contains one grid cell triple (x,y,z).

Command: **griddiff <binary-grid> <ASCII-grid> [<significant digits>]**

see also: GRIDASCI, ASCIGRID, GRIDSURF, SURFGRID, MF2WASIM

5.1.9 GRIDDIFF

Subtracts the second grid from the first grid of the command line parameters.

Command: **griddiff <ingrid1> <ingrid2> <outgrid> [-f:t|s|d]**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

see also: GRIDASCI, ASCIGRID, GRIDSURF, SURFGRID, MF2WASIM

5.1.10 GRIDEDIT

A basic tool for editing single grid cells. The coordinates of the grid cell has to be entered as row and cell indices. The edited grid overrides the old version after the user was asked for confirmation.

Command: **gridedit <grid>**

5.1.11 GRIDGAUS

Generating normal distributed, uncorrelated random number fields

Command: **gridgaus <outfile> <ncols> <nrows> <xll> <yll> <cellsize> <nodata-value> <average> <std.dev.> <step> [<init_value>]**

The ASCII formatted output grid is specified by its name <outfile>, its dimensions <ncols> and <nrows> (number of columns and rows), the coordinates of the lower left corner <xll> and <yll>, the <cellsize> and the nodata-value. The grid values are generated randomly in space according to the specified command line parameters <average> and <standard deviation> of the normal probability distribution. The values are spatially uncorrelated. In order to get reproducible results, the random number generator may be initialized using the option init_value=<init_value> with <init_value> as any real number. The parameter <step> can be used if the results should be integer values. The program generates only integer multiples of <step>. Continuous values will be generated if <step> is set to -1 or no <step> parameter is given (in this case, also <init_value> has **not** to be specified!). Integer cell values may be needed e.g. for discrete, normal distributed values.

see also: GRIDRAND, FIELDGEN, MAKEGRID, ASCIGRID

5.1.12 GRIDGRID

Multiplying two grids on cell by cell basis. If one cell in one or both of the inputs grids is nodata, then the matching cell in the result grid will be also nodata.

Command: **gridgrid <ingrid1> <ingrid2> <outgrid> [-f:t|s|d]**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

see also: GRIDADD, GRIDDIFF, GRIDMULT, GRIDQUOT

5.1.13 GRIDMASK

A mask grid is used to mask an input grid. Only those grid cells are copied into the output grid which have valid matching grid cells in the mask grid (similar to the "setwindow" command in Arc/Info). The cell sizes of the input grid and the mask grid must be identical. The coordinates of the lower left corner must be identical or shifted by integer multiples of the grid cell size. The program can be used to generate the required matching grids for WaSiM from different data sources.

Command: **gridmask <ingrid> <mask-grid> <outgrid> [-f:t|s|d]**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

see also: RESIZE, REMAP, RESAMPLE, REFINE

5.1.14 GRIDMULT

Multiplies all cell values of a grid with a constant value. The constant is typed in interactively during the program run. nodata values in the input grid remain nodata values in the output grid.

Command: **gridmult <ingrid> <outgrid> [-f:t|s|d]**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

see also: GRIDADD, GRIDQUOT, GRIDDIFF, GRIDGRID

5.1.15 GRIDQUOT

Divides all cells of the first grid by the corresponding values of the second grid if both cells are valid. If the second grid contains a cell with the value 0, the operation is not performed there and a nodata value is written to the result grid.

Command: **gridquot <ingrid1> <ingrid2> <outgrid> [-f:t|s|d]**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

see also: GRIDADD, GRIDQUOT, GRIDDIFF, GRIDGRID

5.1.16 GRIDRAND

Generating equally distributed uncorrelated random fields.

Command: **gridgaus <outfile> <ncols> <nrows> <xll> <yll> <cellsize> <nodata-value> <minimum> <maximum> <step> [<init_value>]**

The ASCII formatted output grid is specified by its name <outfile>, its dimensions <ncols> and <nrows> (number of columns and rows), the coordinates of the lower left corner <xll> and <yll>, the <cellsize> and the nodata-value. The randomly generated and grid values are equally distributed between the parameters <minimum> and <maximum>. The values are spatially uncorrelated. In order to get reproducible results, the random number generator may be initialized using the option init_value=<init_value> with <init_value> as any real number. The parameter <step> can be used if the results should be integer values. The program generates only integer multiples of <step>. Continuous values will be generated if <step> is set to -1 or no <step> parameter is given (in this case, also <init_value> has **not** to be specified!). Integer cell values may be needed e.g. for discrete values like land use classes.

see also: GRIDGAUS, FIELDGEN, MAKEGRID, ASCIGRID

5.1.17 GRIDSTAT

Prints the data contained in the 12 records of the grid header to the screen: number of rows, number of columns, x- and y-coordinates of the lower left grid corner, cell size, nodata value, and the statistic parameters, which are the number, sum, minimum, maximum, average and standard deviation of all valid cells within the grid. The option "full" forces the program to actually recalculate the statistics.

Command: **`gridstat <ingrid> [full]`**

5.1.18 GRIDSURF

Converts a binary grid into an ASCII format which can by read by the GS Surfer software. This format uses a reversed order of rows (from bottom to top) and a different header compared to WaSiM and Arc/Info.

Command: **`gridsurf <wasim-ingrid> <surfer-outgrid>`**

see also: SURFGRID, ASCIGRID, GRIDASCI, MF2WASIM, GRIDCOLM

5.1.19 MAKEGRID

Generating a grid with defined coordinates and structure and initializing all cells with a unique initialization value. The resulting is an ASCII formatted grid which can be converted into a binary grid using e.g. ASCIGRID.

Command: **`makegrid <grid name> <ncols> <nrows> <xll> <yll> <cellsize> <nodata-value> <value> [-f:t|s|d]`**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

see also: ASCIGRID, FIELDGEN, GRIDRAND, GRIDGAUS

5.1.20 MF2WASIM

Converting MODFLOW-grids into WaSiM-grids. Since MODFLOW does not store the grid structure, the parameters **`<xll>`**, **`<yll>`**, **`<cellsize>`** and **`<nodata-value>`** has to be specified interactively during the program run.

Command: **`gridsurf <modflow-ingrid> <wasim-outgrid>`**

see also: SURFGRID, ASCIGRID, GRIDASCI, GRIDSURF, GRIDCOLM

5.1.21 RECLASS

Classifies the data of a grid into classes which are written to the output grid. The class number is the integer fraction of the quotient of the value of the input grid minus an offset and the class width. Both parameters, the offset and the class width, are specified interactively by the user. If, for instance, the class width is set to 100 and the offset is set to 0, RECLASS can be used to generate a grid containing elevation belts from the digital elevation model, where the class number is the first digit of the 100 m – elevation belt.

Command: **`reclass <ingrid> <outgrid> [-f:t|s|d]`**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

5.1.22 REFINE

REFINE is a powerful tool to change the grid properties like the cell size and the lower left coordinates, thus the tool can be used to synchronize two or more grids with respect to their coordinates and cell sizes. If any changes in the cell size and the coordinate origin of the grid are done, an interpolation is required. There are 3 methods for choice, of which each has specific use cases:

1. Nearest-neighbor-method: For getting the value in the output grid, the nearest matching cell of

the input grid is looked for and taken as result value. This method should be used if categorized data like land use or soil types should be converted into a finer grid resolution.

2. single bi-cubic interpolation: for each grid cell of the output grid a parabolic surface is constructed using that 3 x 3 window of the input grid which is nearest to the position of the actual output grid cell. The equation for the parabolic surface is given in the description of the program TANALYS (equation 143 and 143). The output value *z* is get by putting the actual x- and y-coordinates in equation 142. This method is fast and gives smooth surfaces, it can be used for continuous data like elevation models, slopes, aspects etc., but not for categorized data, which are not allowed to interpolate.
3. multiple bi-cubic interpolation: For getting even smoother transitions from cell to cell, not only one but up to four parabolic surfaces are constructed after equation (5.1.1) . The 3 x 3 windows are centered on the four grid cells which are nearest to the actual output cell. If the output cell is exactly on a link between two input cells, then only two parabolic surfaces are constructed. The results of the four interpolations after (5.1.1) are then weighted using the inverse distance weighting interpolation with distances depending on the location on the central cells of the up to four parabolic surfaces relative to the output cell location. This method is usually even more accurate than the single bi-cubic interpolation but like those method it cannot be used for categorized data (e.g. land use).

Command: refine **<ingrid> <outgrid> [<method(1, 2 or 3>] [<xll>] [<yll>]**
<method> is one of the above described methods for interpolation. <xll> and <yll> are the new coordinates of the lower left corner of the output grid. The last three parameters are optional, the default values are method 3 and the coordinates of the input grid.

The program asks interactively for the new cell size, which may be any size. The new number of rows and columns is calculated considering also new origin coordinates. If the output grid cell size is much larger then the input grid cell size it is recommended to use the program RESAMPLE or a combination of RESAMPLE and REFINE to get the best results. The method for generating aggregated values used in RESAMPLE gives usually better results when converting a grid into a coarser raster. If, for example, a 50 m grid should be converted into a 70 m grid, it should be converted in a first step into a 35 m grid and the an aggregation using RESAMPLE should be performed to get the 70 m grid.

5.1.23 REMAP

Changes the values of a grid according to a remap (or translation) table. The remap table contains ranges of input values which are replaced by a single output value.

All data of the input grid within the interval]0,100] (lower limit is not included in the interval) are converted to 1 in the output grid. Similarly, all values within the interval]100,101] are replaced by the output value 2 etc. Data, which are not included in any of the given intervals in the remap table are copied into the output grid without changes. **Note:** Also nodata-values are handled like normal numbers, so REMAP can also be used to change the nodata code or to blank some regions (e.g. to hide a subbasin).

example for a remap-table:

```
0   100     : 1
100         101     : 2
101         104.5   : 3
110         128     : 7
```

Command: **remap <ingrid> <remap-table> <outgrid> [-f:t|s|d]**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

5.1.24 RESAMPLE

Re-samples the ingrid to an outgrid with a larger cell size which has to be an integer multiple of the input cell size (example: a 25 m grid can only be re-sampled into a 50 m-, 75 m-, 100 m, 125 m-grid etc.). The coordinates of the grid origin remain unchanged, the numbers of rows and columns is changed according to the new cell-size. The user has the choice between 3 methods: the output cell will either contain (1) an arithmetic average of all corresponding input cell (at least 50 % of the resulting grid cell area must be covered by valid input data cells) → recommended for continuous data, or (2) the nearest neighbor in the input grid to the corresponding center of the output cell → recommended for some derived continuous data like aspect angle and also for categorized data, or (3) the most frequent value of all corresponding input grid cells → recommended for some use cases of categorized data like land use data.

Command: **`resample <ingrid> <outgrid> [-f:t|s|d]`**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

5.1.25 RESIZE

RESIZE can be used for cutting out a specified area of the input grid. The new grid has usually different coordinates and row and column numbers than the ingrid. The cutting area is specified interactively.

Command: **`resize <ingrid> <outgrid> [-f:t|s|d]`**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

5.1.26 SURFGRID

Converting a SURFER-ASCII-grid into a binary WaSiM grid. Beside the different headers, the SURFER grid format has a reversed order of rows compared to WaSiM or Arc/Info.

Command: **`gridsurf <surfer-ingrid> <WaSiM-outgrid>`**

see also: GRIDSURF, ASCIGRID, GRIDASCI, MF2WASIM, GRIDCOLM

5.1.27 TANALYS

Command: tanalys **`<control-file> [-f:t|s|d]`**

Output format: t = ASCII grid, s = single precision binary, d = double precision binary

One of the most important data sets for using the hydrologic model WaSiM is a digital elevation model (DEM), especially when applying the model to mountainous regions. The DEM can be used to calculate a number of derived data sets, like local slope, aspect and curvature. But also some important hydrologic information can be determined by using these derived information. This are, for example, flow directions, flow accumulations (specific catchment area), the river network and the subbasin structure. Its also possible to calculate topographic shading and sky view factors for parameterizing radiation sensitive models. The program TANALYS performs a complex analysis of the digital elevation model. In a series of steps, the data sets as shown in figure 5.1.27.1 are generated. Arrows are indicators that a data set can only be generated if the arrow source already exists. The topographic analysis is controlled by a control file which uses all the features of the WaSiM control file. Some parameters controlling the extend of the analysis and specifying some threshold values or constants and a number of file names have to be specified in this control file.

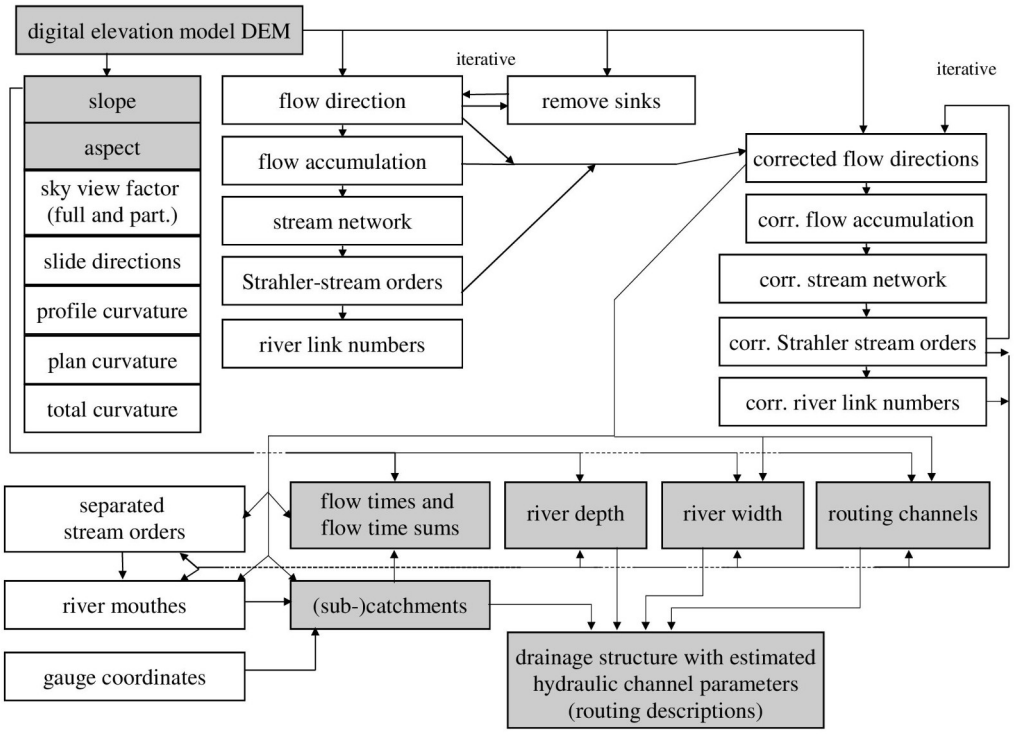


figure 5.1.27.1: topographic analysis of a digital elevation model by TANALYS

Only the shaded data sets are really needed by WaSiM. The grid containing the (sub-)catchments and the text file containing the routing descriptions are required only if the infiltration- and soil module and the runoff routing should be modeled, aspect and slope are required only for the radiation correction and temperature modification (and for the generation of interflow in model version 2) and the will be generated internally during the WaSiM initialization if they are not already there. The flow time sums are used for routing surface runoff and interflow, the grid is not absolutely necessary. River width and depth and the routing channel codes are required, if the model version 2 should consider interactions of surface water and groundwater (infiltration and exfiltration).

The basis of the topographic analysis is the digital elevation model (DEM). Slope, aspect and curvatures are determined using the algorithms given by e.g. BINLEY and BEVEN (1993) or TARBOTON et al. (1993). A 3 x 3 moving window is used to construct a 2nd-order parabolic surface fitting the 9 points. For each of the 9 points equation (5.1.1) holds:

$$z = Ax^2y^2 + Bx^2y + Cxy^2 + Dx^2 + Ey^2 + Fxy + Gx + Hy + I \qquad \textbf{(5.1.1)}$$

If local coordinates are used with 1,1 in the upper left cell and if the distance between the points is normalized and set to L=1 (which is allowed for quadratic grid cells), it is easy to get a solution of the linear equation system (coordinates x, y ∈ {1, 2, 3}). The coefficients A to I are then given by:

$$I = z_{2,2}$$
$$H = (z_{2,1} - z_{2,3})/(2\mathrm{L})$$
$$G = (z_{3,2} - z_{1,2})/(2\mathrm{L})$$
$$F = (-z_{1,1} + z_{3,1} + z_{1,3} - z_{3,3})/(4\mathrm{L}^2)$$
$$E = [(z_{2,1} + z_{2,3})/2 - z_{2,2}]/L^2$$
$$D = [(z_{1,2} + z_{3,2})/2 - z_{2,2}]/L^2$$
$$C = [(-z_{1,1} - z_{1,3} + z_{3,1} + z_{3,3})/4 + (z_{1,2} - z_{3,2})/2]/L^3$$
$$B = [(z_{1,1} + z_{3,1} - z_{1,3} - z_{3,3})/4 - (z_{2,1} - z_{2,3})/2]/L^3$$
$$A = [(z_{1,1} + z_{1,3} + z_{3,1} + z_{3,3})/4 - (z_{1,2} + z_{2,1} + z_{2,3} + z_{3,2})/2 + z_{2,2}]/L^4$$

	x=1	x=2	x=3
y=1	$z_{1,1}$	$z_{1,2}$	$z_{1,3}$
y=2	$z_{2,1}$	$z_{2,2}$	$z_{2,3}$
y=3	$z_{3,1}$	$z_{3,2}$	$z_{3,3}$

(5.1.2)

Some local parameters can by derived directly from this parameters by:

slope angle $\hat{\beta} = \arctan(\sqrt{G^2 + H^2})$ **(5.1.3)**

aspect angle $\hat{\Omega} = 180 - \arctan(H/G) + 90(G/|G|)$ **(5.1.4)**

profile curvature $\Phi = -2(DG^2 + EH^2 + FGH)/(G^2 + H^2)$ **(5.1.5)**

plan curvature $\omega = 2(DH^2 + EG^2 - FGH)/(G^2 + H^2)$ **(5.1.6)**

total curvature $\chi = \omega - \Phi = 2E + 2D$ **(5.1.7)**

The sky view factor may be of interest if spatially distributed radiation budgets are required for modeling the influence of topography on hydrological processes. Using the algorithms as shown in Chapter 1.3 for calculating the shading of a DEM, for each grid cell all directions (in steps of e.g. 10° starting with north) are scanned for that sun angle which would lead to shading by surrounding hills or mountains. Slope and aspect of the actual cell are also taken into consideration. For each segment, the result is an horizon overhead angle $\beta_{h,i}$ which is used to calculate the fraction of diffuse incoming shortwave radiation for this segment *i* of the sky relative to the theoretical diffuse incoming shortwave radiation of the same segment if there would be no horizon overhead. After OKE (1987), the sky view factor ψ_{sky} can be calculated considering both, LAMBERT's law of the intensity dependency of the radiation on its incoming angle as well as the visible fraction of the sky which decreases with increasing horizon overhead after 1-sinβ_h:

$$\psi_{sky} = \sum_{i=1}^{n} \cos^2 \beta_{h,i} \qquad \textbf{(5.1.8)}$$

with $\beta_{h,i}$ horizon overhead angle for segment *I* [rad]
n number of segments to build a full circle

The shortwave radiation budget R_K can be expressed following eq. (13) as follows:

$$R_K = \frac{R_{sun} \cdot \cos\hat{\Theta}}{\cos Z} + D \cdot \psi_{sky} + RG(1 - \psi_{sky}) \cdot \alpha \qquad \textbf{(5.1.9)}$$

with D diffuse shortwave radiation from the sky [Wh/m²]
R_{sun}, RG direct sun radiation resp. global radiation (direct+diffuse)[Wh/m²]
α mean albedo of the terrain visible from the actual point [-]

The mean value of the sky view factors of all segments is the output value for the sky vie factor grid. As mentioned, this includes LAMBERT's law as well as the nonlinear decrease of the visible sky area and is thus valid for units of diffuse radiation only. If on the other hand only the visible fraction A_s of the sky should be calculated, equation (5.1.8) should be reformulated, using the formula for the surface of a half sphere $A = 2\pi r^2(1\text{-}sin\ \beta_h)$:

$$A_s = 1 - \sum_{i=1}^{n} \sin \beta_{h,i} \qquad \textbf{(5.1.10)}$$

The difference between ψ_{sky} and A_s may be considerable, especially for small and medium horizon overhead angles. This is because equation (5.1.10) does not consider LAMBERT's law of the decrease of the irradiation with an decreasing incoming angle. Because of the squared cosine in equation (5.1.8) small overhead angles does not make much effect at all. Thus, the sky view factors are above 0.99 for $\beta_h < 5°$, whereas at the same time the visible sky fraction decreases down to only 91 %. Figure 5.1.27.2 shows these relations. WaSiM does not use the sky view factor actually but it is planned for future versions to incorporate the effect of horizon overhead on diffuse radiation.

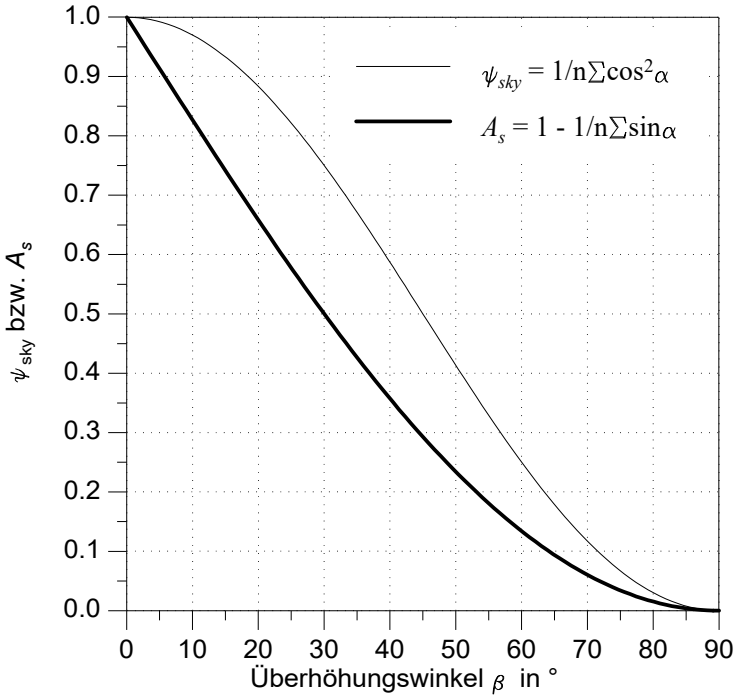


figure 5.1.27.2: ky-view-factor ψ sky and fraction of the sky As depending on the horizon overhead angle □ h

Flow directions are determined by the steepest of the slopes to the at maximum 8 neighboring cells. Flow is allowed into only *one* of these cells. Artifacts like sinks or flat areas which may be the result of generating the elevation model, but also real, natural sinks, e.g. in karst regions, are filled iteratively. Sinks are removed by looking for the lowest point of all points outside but connected to the border of the sink which have a higher elevation than the sink itself. This elevation is then used as replace value for all cells within the sink. If after the filling all boundary cells of the sink, which has grown by at least one cell, still have no gradient to the outside of the sink, the filling is done again, now looking for the next higher elevation of the surrounding cells. This is done iteratively until at least one cell of the (usually now much larger) sink has a flow gradient to the sinks outer side or until the growing sink reaches a nodata value or the boundary of the grid, which both are handled as pour points. Then, the sink has become a "flat region". Now, the flow directions within the flat regions are directed to the one ore more pour points in an iterative way starting at the pour points and looking for undefined flow directions in their neighborhood.

Profile and plan curvature may be used for determining zones of convergence or divergence. This can be used for generating multiple flow directions by weighting the flow fractions by flow lengths and slope. WaSiM does not use neither the profile nor the plan curvature.

After calculating the flow directions and filling the sinks, the flow accumulation can be calculated, which is the catchment area for each grid cell in units of grid cells draining directly or indirectly

through each cell. The accumulation is done by following the flow directions of each cell until the grid boundary or a nodata value have been reached and summing up the numbers of touched cells. The result value includes the cell itself, so the cells at the catchment boundary have already a flow accumulation value of 1 in order to avoid errors e.g. in calculating the topographic index. For calculating the topographic index, the flow accumulation value should be scaled by the cell size, in order to get grid scale independent results.

By setting a threshold for the flow accumulation, the river network can be extracted as a first approximation. The extraction of river networks is a rather complicated task, see e.g. Tarboton et al., 1993. The river network is used together with the flow direction grid to determine the flow orders after STRAHLER and to identify each river link with a river link number.

If the extracted stream network still shows artifacts like parallel rivers in neighboring grid cells, these artifacts can be removed by applying a flow direction correction algorithm. It is also an iterative task. In a first step rivers of STRAHLER-order 1 are checked (head waters). If they are flowing alongside a river of higher order or alongside a river of the same order but with a larger catchment and if there is at maximum 1 no-river grid cells between the rivers (which must have an elevation less or equal to the lower of the river cells), then the smaller rivers flow direction is redirected by 45° to the larger river. After processing the entire grid, the same procedure is done for rivers of STRAHLER-order 2 and then for rivers of STRAHLER-order 3. Then, the correction is done again for rivers of STRAHLER-order 1 resp. 2 in order to correct new artifacts which may have been introduced by the correction of higher order flow directions. After each single correction step a recalculation of the flow accumulations, the river network and the Strahler-orders has to be performed.

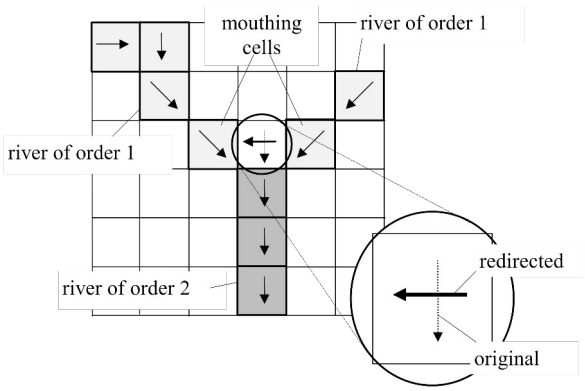


figure 5.1.27.3: flow direction correction at mouths

A further correction may be necessary if grid cells drain between the mouthing points of two rivers directly into the higher order river. Then, a redirection of flow directions has to be performed. Figure 5.1.27.3 shows the principle of this correction. Without this correction, the subbasin for the river of order 2 would extend upstream between the catchments of the mouthing rivers of order 1. This is an artifact and it would stand against all logic and against the reality. Therefore the flow direction of the last in-between-grid-cell is redirected to the last cell of the nearest or, if both mouthing rivers have the same distance to the redirected cell, to the larger river.

By separating rivers of order 2 and higher or order 3 and higher and so on, the stream network can be limited to rivers of a given minimum order (or corresponding catchment area). The separated rivers are newly numbered and then the mouths are determined, where these rivers are flowing in higher order rivers. Mouthing points are the last grid cells of a river when looked on it in flow direction.

For each stream link of the river number grid a flow velocity after MANNING-STRICKLER is estimated.

The velocity depends mainly on the mean slope as given by the length and the elevation range of the link.

$$v = M \cdot R_h^{2/3} \cdot I^{1/2} \quad \textbf{(5.1.11)}$$

$$\dot{V} = M \cdot A \cdot R_h^{2/3} \cdot I^{1/2} \quad \textbf{(5.1.12)}$$

with v flow velocity [m/s]
M roughness parameter after Manning-Strickler (10 … 40) [$m^{1/3}/s$]
A cross section area of the flow (rectangle profile with depth/width = 1/8)[m^2]
R_h hydraulic radius (= flow cross section area A / channel cross section perimeter U) [m]
I average slope of the river link [m/m]

Beside the roughness parameter M, which is specified as a global parameter in the TANALYS control file, also the hydraulic radius R_h in equation (5.1.11) is an unknown parameter which is estimated automatically by TANALYS. Since R_h also depends the absolute flow, another global parameter for the specific discharge q [liter/(s·km^2)] has to be specified in the control file. The absolute discharge at a given profile is then the product of the local catchment area A_E and the specific discharge q. Then it is assumed that the channels have rectangle profiles with a depth to width ratio of 1:8. The hydraulic radius R_H is then given by:

$$R_h = \frac{A}{1.25\sqrt{8A}} \quad \textbf{(5.1.13)}$$

Since A depends on the flow velocity, which should be the result of the calculation, an iteration is required. The first assumption is v = 1 m/s. After calculating the first hydraulic radius R_H, the second estimation of v is get after equation (5.1.11) , using the constant parameters M and I and the first estimation for R_H. The iteration converges to some certain values for v and R_H and thus for the river depth and width. Convergence is assumed for iteration errors of $\Delta v < 0.05$ m/s or after 12 iteration steps. The calculated flow velocities are valid only for the specified specific discharge q and for the roughness parameter M. The velocity is converted into flow times the water needs to cross each grid cell of the actual river link, see equations (5.1.15) and (5.1.16) . In order to get realistic results for the discharge routing in later WaSiM applications, the specified specific discharge should be in the range of the specific medium flood discharge. These are the discharges which are most important for the routing algorithm.

Also for grid cells which are not river cells a flow velocity is calculated. This surface runoff velocity is also calculated after equation (5.1.11) . The flow velocity is converted into the flow time required for crossing the grid cell and summed up along the flow path for each grid cell until the basin outlet is reached (including river cells) in order to generate a flow travel time grid. This grid is used by WaSiM for routing the surface runoff as well as the interflow within subbasins. As a simplification of the hydraulic calculations, a flow depth of R_H = 1 cm is assumed for surface runoff. Also a constant Manning-roughness parameter of M = 10 $m^{1/3} \cdot s^{-1}$ is presumed. Then, the flow velocity on land surfaces depends only on the local slope, see equation (5.1.14) . The flow time t the water needs for crossing a cell, depends on the effective flow length (diagonal or perpendicular to a cell side):

$$v = 10 m^{1/3} s^{-1} \cdot (0.01\,m)^{2/3} \cdot \sqrt{I} \quad \textbf{(5.1.14)}$$

$$t = \frac{cs \cdot \sqrt{2}}{v} \qquad \textit{for flow directions} \nearrow\searrow\swarrow\nwarrow \quad \textbf{(5.1.15)}$$

$$t = \frac{cs}{v} \qquad \textit{for flow directions} \uparrow\rightarrow\downarrow\leftarrow \quad \textbf{(5.1.16)}$$

Sub-catchments are delineated using the mouthing points or any other pour points given in separate

grids, e.g. locations of gauging stations as numbered grid cell codes in a pour point grid. The flow times for crossing the grid cells are now also summed up for each cell until the next pour point is reached. Grid cells which are part of a routing channel are coded with the number of the largest of its tributaries. This will be the so called link-grid, which is used in WaSiM-version 2 for budgeting the re-infiltration rates with the flow in the rivers. River width and depth are also written into separate grids. At the same time, the drainage structure is analyzed and written into a text file which contains also the estimated hydraulic parameters. The structure matches the requirements of the routing descriptions in the section [routing_model] of the WaSiM control file. It can simply be inserted into this file.

Figure 5.1.27.4 shows an example for a flow travel time grid. The catchment shown is the Thur basin, subdivided into 12 subcatchments, each of which is observed by a flow gauging station at the outlet. The separate zones are flow times in hours the water needs to reach the next subbasin. Bright areas stand for short dimes, dark areas means that the water needs much more time to reach the outlet, up to 13 hours in the example. The flow time spectrum depends on the MANNING-roughness parameter as well as on the presumed flow depth for overland flow and on the presumed specific discharge for the channels

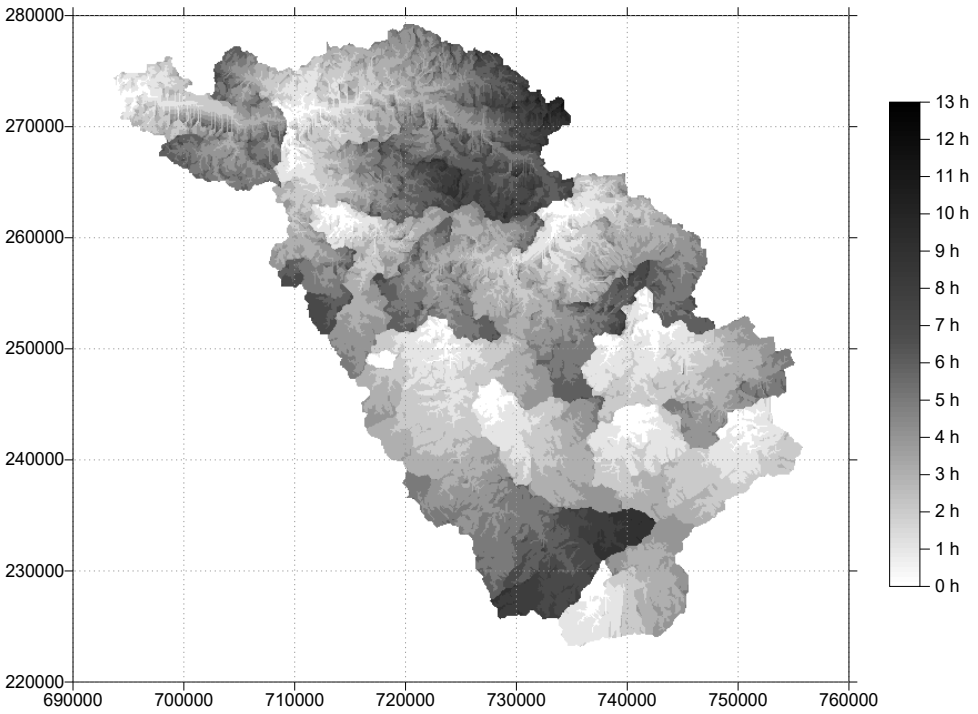


figure 5.1.27.4: flow travel time grid for the river gauge network of the Thur-basin, hour zones

5.1.28 TOPOFACT

Calculates the soil-topographic index $\ln(A/(T_0 \cdot \tan\beta))$ using the local catchment area A, the local transmissivity T_0 and the local slope angle β. In opposition to the usual definition of the soil topographic index, A is the local catchment, not the local specific catchment (this is a program restriction). In order to compare TOPOFACT´s results to the topographic indices of other programs, the value ln(cellsize) has to be subtracted from all grid cells. If only the topographic index without the influence of the transmissivity is wanted, then for T_0 a grid with constant cell values of 1.0 should be used.

Command: `topofact <flow accumulation> <transmissivity> <slope> <outgrid> [-f:t|s|d]`
Output format: t = ASCII grid, s = single precision binary, d = double precision binary

5.1.29 ZONESTAT

Calculates the statistics of the values in an input grid separately for each zone of a zone grid. The parameters are: zone ID, zone value, minimum, maximum, sum, average, standard deviation.

Command : `zonestat <ingrid> <zone grid> <statistic file>`

Each zone is coded by an integer value (0<zone vale<32768) in the tone grid. Zones doe not have to cover contiguous areas, e.g. if elevation belts should be used as zones. Such zones can be generated using the programs RECLASS or REMAP.

The output file contains the statistic parameters of the input grid separated after the zones of the zone grid in an ASCII text file.

5.2 Working with station data (meteorological time series)

5.2.1 IDWP

Performs a temporally and spatially interpolation of precipitation data in order to get daily courses of precipitation data for climate stations which data are given as 12-hour or 24-hour precipitation sums. Required input data are both, the high resolution hourly precipitation data as well as the 12-hour and/or 24 hour low resolution precipitation data of as much stations as available. The precipitation of the high resolution data are interpolated to the locations of the low resolution data. Then, the results are weighted on the total sums of these low resolution data according to the matching period. This process is described in detail in the following paragraphs.

The interpolation is done in two steps. During the first step, the precipitation of the high resolution data is interpolated to the location of the conventional climate stations using inverse distance weighting interpolation (IDW). Some parameters have to be entered by the user, like the power of the inverted distance p in equation (4) and the maximum search radius for valid station data around the location of interpolation d_{max}. Furthermore, it is possible to specify fix upper and lower limits which should not be exceeded by the results. If this should happen such unwanted results are replaced by user specified default values, e.g. for precipitation values below 0.1 $mm \cdot h^{-1}$, which may be replaced by 0 mm in order to avoid too much and too long precipitation events due to the interpolation algorithm.

For each grid cell, all stations within the specified search radius around that cell are used for the interpolation. The weight parameter is used as exponent to calculate the power of the distance, which is then inverted in order to give more far stations a fare less weighting for the interpolation result. The sum of all inverse weighted distances is scaled to 1.0, the resulting scaling factor is then applied to each station weight and finally multiplied by the station data. The sum of all station data is the interpolation result.

After this interpolation of hourly values (7:00 to 7:00 of the next day) for the locations of the conventional climate stations, the 12-hour or 24-hour sums of the interpolated values are scaled to the observed sums, the resulting scaling factor is applied to each interpolated hourly value of the matching 12-hour resp. 24-hour interval.

The output file contains the original (e.g. 1-hour-values) values of the high resolution stations and the interpolated data of the conventional climate stations in the same temporal resolution. It can be used directly as an input file for the interpolation algorithms in WaSiM. The hourly input data have to be read in IDWP in 1/10 mm, the 12-hour and 24-hour data in mm. The result is given in 1/10 mm/h. The file names and parameters are specified interactively by the user or by an ASCII-file via redirected input.

5.2.2 REGRESS

This program is a GUI program for DOS. It will not run under Windows Vista/7 and above. To run it, the freeware software DOS-box can be used, which is simulating a DOS machine in a normal window.

Regress calculates altitude dependent gradients for input data of any time steps. The regression is done for two separate altitude ranges which can be specified by the user. If the regressions does not cross each other in a given altitude range, the profile may consist of up to three sections.

If the input data are not only consist of high resolution data (e.g. hourly values) but *also* of data of conventional stations, which may be represented by a number of observations at given times of the day, these data are also included in the regression. This integration is done in a second order temporal interpolation: for the times with valid data of both input types (high resolution *and* conventional observations) two regressions are calculated, one for the high resolution data only, the other for all data including the conventional observations. This is done for each time when all data are available. The deviations between both regressions are then temporally interpolated between the observation times. Using the interpolated deviations, the algorithms estimates "observation" data for the conventional climate stations for the unobserved times. The last step is the re-calculation of the altitudinal regressions using now all data, including the estimated data for the conventional stations in-between the observation times.

The gradients can be viewed on the screen and also be sent to a HP-Desk-jet printer (as raw bitmap) or to a HPGL-plotter in HPGL-format. This can be done by typing <p> during the display of the gradients. This HPGL-files can usually also be printed on most Postscript printers, also there is a freeware called printgl available for printing such files on other windows printers.

The parameters for REGRESS are saved in a parameter file. This file is also used by REGR, the non interactive version of REGRESS (if all the parameters were optimized, no graphical user interface is necessary). However, it is highly recommended to use REGRESS for at least a part of the input data in order to be sure to specify the right parameters, especially the right inversion altitudes. REGRESS generates an ASCII-result file (and some ASCII-files containing efficiency criteria and the original and temporally interpolated input data). The regression file contains for each time step a set of parameters for 3 linear equations for the at maximum 3 valid ranges and 3 parameters for the areal regression of the residuals of the altitudinal regressions (see also description of the regression in chapter 2.3.2).

The maximum number of stations to be considered is 200. For each station, a 0/1 code in the last row of the control file (interactively checked in REGRESS after specifying all parameters) specifies the use of the station for the regression.

Command: **`regress <parameter_file>`**

5.2.3 REGR

REGR is the non interactive version of REGRESS for use under Windows or Linux or for use with already well parameterized data sets, i.e., the inversion altitudes and other parameters are known, e.g. from former runs of REGRESS. REGR uses the same parameter file like REGRESS but generates a binary output file instead of an ASCII file. Both output file versions can be read by WaSiM without conversions, but the binary version is smaller and faster to read but cannot be transported onto other platforms. WaSiM recognizes the format of the regression result file automatically by the first 2 words "jr" and "mo", separated by only one space or tab stops. The following programs can be used for converting the file formats into another.

Command: **`regr <parameter_file>`**

5.2.4 REGRASCI

Converts a binary regression as generated by REGR file into an ASCII-regression file as generated by REGRESS. This may be wanted for file transfer between different systems (e.g. DOS/WINDOWS ↔ UNIX) or for looking into the file for data checking. Since WaSiM recognizes the format of the regression file automatically, a re-converting into the target systems binary format is not necessary but may be performed in order to save hard disk space. The specification of both formats are described in chapter 3.2.2.

Command: **regrasci <binary_in_regression_file> <ASCII_out_regression_file>**

5.2.5 ASCIREGR

The opposite of REGRASCI, converts ASCII-regression files into binary regression files.

Command: **asciregr <ASCII_in_regression_file> <binary_out_regression_file>**

5.2.6 GMDtoTAB

This program should be used to convert so called GMD formatted data into the WaSiM table format as described in chapter 3.2.2. Data of a number of input files are copied into one output file using parameters as given in a control file like the time period, the station names and codes and scaling factors. The GMD-format allows a discontinuous storing of data, this is not allowed, if GMDTOTAB should be used for conversions, so all data have to be in a strong temporal order without gaps or multiple occurrences.

Command: **gmdtotab <control file>**

The control file is structured as follows:

1st row: number *n* of following GMD-data files
2nd row: start time of conversion as YYYY MM DD hh mm
3rd row: end time of conversion as YYYY MM DD hh mm
4th row: <name of the 1st GMD-file> <station code><scaling factor>
5th row: <name of the 2nd GMD-file> <station code><scaling factor>
.
. (etc. for all GMD-input files)
.
last row: <name of the WaSiM output file>

5.2.7 QtoSPEND

Converts discharges in m^3/s into specific discharges in mm/time step.

Command: **qtospend <discharge_file> <specific_discharge> [<header_rows>] [<number of data columns>] [<time step in hours>]**

The number of header rows is 5 by default, the number of data columns is determined automatically. If both values are specified explicitly, no automatic check is done. If the time step differs from 1 hour, then the time step should be specified in hours, e.g. in order to get mm/day if using 1-day-resolution input data.

5.2.8 SPENDINV

Converting specific discharges in mm/time step into time steps/mm. This file may be used for fitting regression parameters to discharge recessions in order estimate the recession parameter *m* for WaSiM version 1 (soil model). The number of header rows should be 5, the number of data columns is determined automatically.

Command: **spendinv <specific discharges> <inverted specific discharges>**

5.2.9 SONNEREL

Converting absolute sunshine duration into relative duration. The parameters for the geographic location (latitude, longitude), for time steps of input and output data and for a time shift (e.g., if the official time does not match the mean local sun time) must be specified interactively. The program calculates the on the basis of these parameters for each time step a theoretical maximum sunshine duration which is used for scaling the absolute sunshine duration in the input file into a relative sunshine duration.

Command: `sonnerel <absolute sunshine_duration> <relative_sunshine_duration>`

5.3 Software for visualization

5.3.1 REGRESS

In opposite to REGR, the program REGRESS is able to print the calculated altitudinal gradients for meteorological input data as graphics on the screen of the computer as well as on HPGL-Plotters or printers. Different colors show the different input file regressions (e.g. for hourly and 6-hourly data). By pressing “p” during the display of a graph, this page can be printed or plotted (into a file) in HP-Desk-jet or HPGL-format. For further description see section 4.2 of this documentation.

5.3.2 SHOWGRID

Showgrid is a Windows GUI application which can be purchased by the author. It has it's own documentation. The highlights are:

- Binary and ASCII-formatted WaSiM-Grids can be viewed. Also binary WaSiM-stacks can be viewed
- Multiple grids can be loaded and cycled through in a load list – the number of grids is not limited (they will be loaded from disk each time they are redrawn)
- Zooming in and zooming out is possible
- The levels of stacks can be viewed separately simply by clicking through the levels.
- Grid lines can be viewed as helper
- There are 6 default color palettes defined (each as smooth change from one color to two other colors), the legend is divided into 21 parts.
- It is possible to define user palettes (they will be saved to the users directory and be available in the palette menue as any other palette)

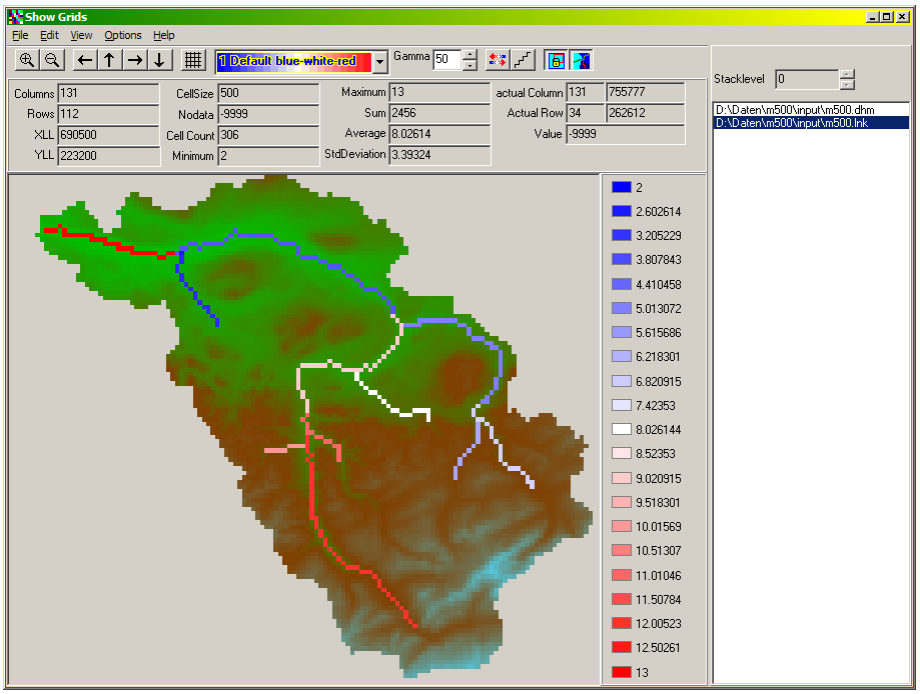


figure 5.3.2.1: screenshot of ShowGrid

- Color rendering can be done continuously or discontinuously (in steps)
- Color palettes can be inverted
- The color saturation of standard palettes can be changed in wide ranges for better viewing results
- Statistic values of each grid (and of each single layer in a stack) will be viewed automatically
- When loading the grid, the statistics will be newly calculated (since ASCII-grids does not contain statistics)
- Single stack layers can be stored as grids
- The actual viewed frame can be copied as bitmap to the clipboard to be inserted into any other Windows application
- Selected cells (by using the mouse, any number of cells can be selected) could be changed. This allows for an easy editing of grids.
- When drawing a “line” with the mouse and pressing Shift+Ctrl+C, the profile of the selected cells is shown in a separate window (which also can be copied to the clipboard)
- The range of the viewed values can be defined manually (default is the automatic calculation of the scale)
- For each grid, configurable arrows can be overlay onto the grid for showing gradients
- If velocity grids are available, these grids can be loaded for viewing the gradients (scaled and oriented by using the input values, e.g. groundwater flow directions and velocities)
- When zooming, the clipping rectangle can be fixed, so other grids will be zoomed into the same region automatically
- Grids can be drawn as overlay over other grids, thus allowing e.g. the projection of the river network on the subbasin grid or the elevation model (see figure 5.3.2.1)
- Showgrid has to be activated with a license (a demo license is available by the author)

- When using Linux, Showgrid can be installed using the wine package, so it is as simple as under Windows to use Showgrid.

5.3.3 GRAPHLINES

Like Showgrid, Graphlines is a Windows GUI program – but not for viewing grids but for viewing statistic files, e.g. WaSiM input tables or WaSiM result statistics. The most important features are:

- Automatic parsing of WaSiM statistic files: each column will be shown as a single graph (a so called series)
- Titles of the columns are taken from the header of the statistics file (e.g. sub-basin code)
- The name of the series will be talen from the header of the input file (and can be edited)
- The number of files to be loaded at the same time is not limited (only the hardware and the 1.5 Gigabyte-limit of 32-bit operating systems may limit the number of files to be loaded (Graphlines runs as a 32 bit application only)

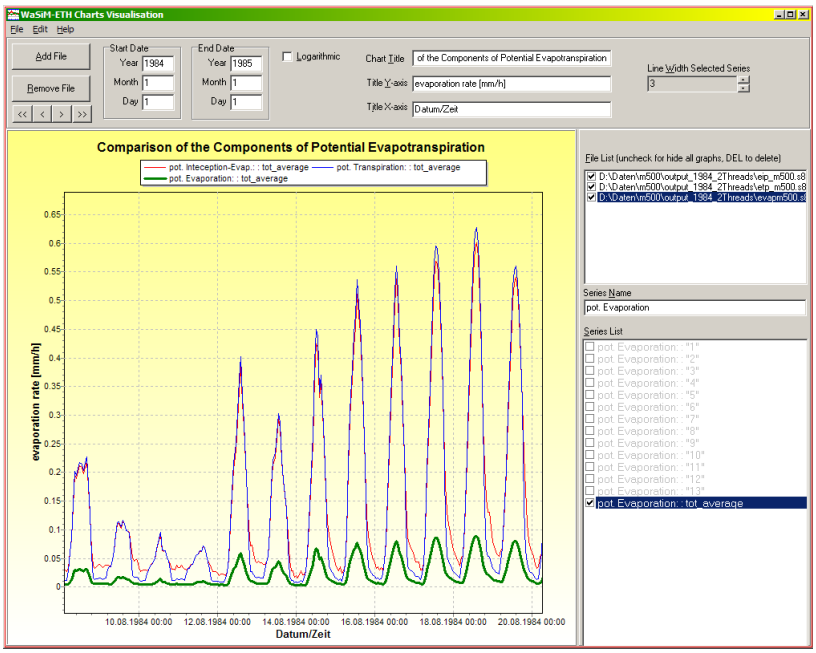


figure 5.3.3.1: Example for Graphlines

- Each file can completely be hide or viewed by a single click on a checkbos
- Each series within a statistics file can also be hide and shown (default: only the last series is drawn when loading the file)
- All selected series will appear in a legend at the top of the chart.
- Files with different temporal resolution can be plotted commonly (Graphlines keeps track of the date in the date/time columns). Thus is it possible to compare model results from different temporal resolutions.
- Each series is drawn in a separate color and/or line thickness. The order of selecting the series for drawing defines the color.
- A double clikc on the series allows to change the color manually
- The currently selected series (in the series list) will be drawn with a bold line (the line thickness can be configured)

- Logarithmic scaling of the ordinate is possible
- One of the most important features: by using the mouse it is very easy to zoom into regions of interest
- Also, the region can be moved by using the right mouse button. Both actions are virtually without time lag, zooming is even done with an animation.
- Like Showgrid, Graphlines needs to be activated by a computer or user specific license. Demo licenses are available by the author.

5.4 Other programs

5.4.1 MAKEGWN

Calculates the gross and net groundwater recharges using the result files of WaSiM version 1 (TOPMODEL-approach).

Command: `makegwn <results_SD> <results_QB> <brut_recharge> <net_recharge> [<header rows>] [<data columns>]`

The input files are the ASCII-result files of a WaSiM (version 1) run containing the average saturation definite of all subbasins (SD) and the base flow (QB). The result files contain the brutto recharge (positive values and negative values) and the netto recharge (only positive values). The specification of the number of header rows and the number of data columns is optional.

5.4.2 RESAGGR

Temporal aggregation of hourly- or daily WaSiM result (or any other files in any WaSiM-table-format). The inputs are summed up to daily and monthly values for each data column of the input file. At the end, a total sum is given. The result files get the same name as the input files except for the first character of the file extension which is replaced by a “t” for daily aggregates and by an “m” for monthly aggregates. The number of header rows and the number of data columns is detected automatically but can also be specified explicitly.

Command: `resaggr <input_file> [<number of header rows>] [<number of data columns>]`

5.4.3 RESAGG

Aggregates the input data for a given number of time steps. The name of the result file and the aggregation base (aggregation over how many time steps?) must be specified in the command line. It is a more general version of RESAGGR. The number of header rows and the number of data columns is detected automatically but can also be specified explicitly.

Command: `resagg <input_file> <output_file> <aggregation_time_steps> [<number_of_head_rows>] [<number_of_data_colums>]`

5.4.4 RESMEAN

Averages for each data column the 1-hour or 1-day results of WaSiM to daily and monthly averages. At the end, a total average is given for each zone. The result files get the same names as the input files except for the first character of the file extension which is replaced by a “t” for daily averages and by an “m” for monthly averages. The number of header rows and the number of data columns is detected automatically but can also be specified explicitly.

Command: `resmean <input_file> [<number_of_head_rows>] [<number_of_data_colums>]`

5.4.5 RESMEANI

Averages the input data for any given number of time steps. The name of the result file and the aggregation base (averaging over how many time steps?) must be specified in the command line. It is a more general version of RESMEAN. The number of header rows and the number of data columns is detected automatically but can also be specified explicitly.

Command: `resmeani <input_file> <output_file> <aggregation_time_steps> [<number_of_head_rows>] [<number_of_data_colums>]`

5.4.6 QUANTIL

Estimates the percentiles of monthly of daily aggregates of WaSiM results, e.g. of the aggregation results contained in the RESAGG(R) or RESMEAN(I) results. The data sets may be normal or log-normal distributed (with 2 or 3 parameters, then a lower limit is used).

Command: `quantil <control_file> [NV|LN2|LN3]`

The control file contains the names of all files which have to be evaluated. The result file has the same name like the control file but the extension ".sta". The optional parameters NV, LN2 or LN3 define the type of distribution function to be used for the parameter estimation. LN2 is the default. If LN3 is chosen, a lower limit has to be specified interactively (using the same units like the input data, e.g. mm/month if input data are monthly aggregated discharge sums). The files are processed in the order as given by the control file. All input files has to be structured identically, i.e., the same number of zones and the same applied aggregation method. For each zone or subbasin, a set of parameters is estimated (average, standard deviation) which are used to calculate the 5%, 50% and 95% percentiles. These percentiles and the minimums and maximums of the input data are written to the output file.

5.4.7 LOWFLOW

Extracts the minimum moving average values from the WaSiM discharge results.

Command: `lowflow <input_file> <output_file> <moving average window_width> <data_column> [/date] <header_rows>`

The input file should be a file with daily discharges in the WaSiM result format. If the model run uses hourly values, then the results should be aggregated first using RESAGGR. The moving average window width is the number of time steps the moving average is calculated for. For daily discharges, the window width should be 7, 15 or 30. The data column specifies the zone (subbasin) to be processed. The first 4 columns of the input file contain the date and time, they are no data columns, so the 5th column of the input file is the 1st data column. The option "/date" forces the program to write the date and time also into the output file. The default number of header rows is 3, any other value must be specified explicitly.

5.4.8 FLOODS

Lists all discharges of a discharge file which are above a specified threshold to an output file.

Command: `floods <input_file> <output_file> <threshold> <data_column> [/date] <header_rows>`

Instead of the interval number in LOWFLOW, a threshold must be specified here. Of all exceeding discharges the maximums are written to the output file for each exceeding period. These maximums have to be real maximums, thus the maximum have to be preceded **and** succeeded by at least one value which is less then the maximum itself. This may important at the begin and at the end of the time series, since otherwise e.g. the starting value which may be a value in the midst of a recession could be taken as a maximum instead. At the end of the time series the runoff hydrograph must be below the threshold in order to write the maximum into the output file.

6 References

Anderson, E.A. (1973): National Weather Service river forecast system - snow accumulation and ablation model. *National Oceanographic and Atmospheric Administration (NOAA), Tech. Mem., NWS-HYDRO-17, U.S. Department of Commerce,* Silver Spring, MD. 217 S.

Bahr, D. B., M. F. Meier, and S. D. Peckham (1997), The physical basis of glacier volume-area scaling, *J. Geophys. Res.*, 102(B9), 20,355– 20,362, doi:10.1029/97JB01696.

Bardossy, A. (1994): Regionalisierung von hydrologischen Kenngrössen und Modellen. *Beitrag im 18. Fortbildungslehrgang Hydrologie des DVWK „Niederschlag-Abfluss-Modelle für kleine Einzugsgebiete und ihre Anwendung“*, 19. bis 23. September 1994, Karlsruhe.

Beven, K.J. and Kirkby, M.J. (1979): A physically based variable contributing area model of basin hydrology. *Hydrol. Sci. Bull.,* 24 (1), 43-69.

Binley, A. and Beven, K. (1993): Three Dimensional Modelling of Hillslope Hydrology. *Advances in hydrological processes. John Wiley & Sons Ltd., Chichester.* Orig. publ. in: *Hydrological processes- An International Journal,* 5(1), 1-126 (1991) und 6(3), 253-368 (1992).

Braun, L.N. (1985): Simulation of snowmelt-runoff in lowland and lower alpine regions of Switzerland. *Zürcher Geographische Schriften, ETH Zürich*, 21, 166 S.

Brutsaert, W. (1982): Evaporation into the atmosphere. *Kluwer Academic Publishers, Dordrecht*.

Chen, J., and A. Ohmura (1990), Estimation of alpine glacier water resources and their change since the 1870s, Hydrology in mountainous regions - I. Hydrologic measurements, the water cycle, Proceedings of the two Lausanne Symposia, IAHS Publ. No . 193 , 127–135.

Chmielewski, F.-M., Müller, A., Küchler, W. (2007): Possible impacts of climate change on natural vegetation in Saxony (Germany). Int J Biometeorol, 50: 96-104.

Daanen , R. P. a nd Nieber J. L. (2009): Model for Coupled Liquid Water Flow and Heat Transport with Phase Change in a Snowpack. *Journal of Cold Regions Engineering , Vol. 23, No. 2, DOI: 10.1061/ASCE0887-381X200923:243*

Dyck, S. und Peschke, G. (1989): Grundlagen der Hydrologie. 2. *Aufl., Verlag für Bauwesen, Berlin,* 408 S.

Federer, C.A., Lash, D. (1983): BROOK - a hydrologic simulation model for eastern forests. *Water Resources Research Center, University of New Hampshire*.

Förster, K., Garvelmann, J, Meißl G and Strasser, U. (2018): Modelling forest snow processes with a new version of WaSiM, *Hydrological Sciences Journal, 63:10, 1540-1557, DOI: 10.1080/02626667.2018.1518626,*
download: https://doi.org/10.1080/02626667.2018.1518626

Gurtz, J. (1988): Beitrag zu den hydrologischen Grundlagen für die Gewährleistung der gesellschaftlich notwendigen Mehrfachnutzung der Wasserressourcen. *Dissertation (B) an der TU Dresden, Institut für Hydrologie und Meteorologie*.

Grant, S. A., (2000): Physical and Chemical Factors Affecting Contaminant Hydrology in Cold Environments, *Technical rept. 00-21, 37 Pages, U.S. Army Cold Regions Research And Engineering Lab, Hanover NH, Experimental Engineering Div.*

Grant, S. A. and Sletten, R. S. (2002): Calculating capillary pressures in frozen and ice-free soils below the melting temperature. *Environmental Geology* Volume 42, Numbers 2-3, 130-136, DOI: 10.1007/s00254-001-0482-y

Green, W.H. and Ampt, G.A. (1911): Studies on Soil Physics: I. The flow of air and water trough soils, *Journal of Agricultural Sciences*, 4, 1-24.

Gruber, S. (2007), A mass-conserving fast algorithm to parameterize gravitational transport and deposition using digital elevation models, *Water Resour. Res., 43,W06412, doi:10.1029/2006WR004868.*

Hammond, C., D. Hall, S. Miller and P. Swetik, (1992), "Level I Stability Analysis (LISA) Documentation for Version 2.0," *General Technical Report INT-285, USDA Forest Service Intermountain Research Station.*

Hock, R. (1998): Modeling of Glacier Melt and Discharge. Diss ETH 12430, Verlag Geographisches Institut ETH Zürich, 140 S.

Jansson, P.E., Karlberg, L. (2001): Coupled heat and mass transfer model for soil-plant-atmosphere system. – Division of Land & Water Resources, Depart. of Civil and Environ. Engineering, Royal Institute of Technology, Stockholm. Web document:

ftp://www.lwr.kth.se/CoupModel/coupmanual.pdf

Kudryavtsev, V. A., Garagula, L. S., Kondrat'yeva, K. A. and Melamed V. G. (1974). *Osnovy merzlotnogo prognoza* (in Russian). MGU, 431 pp. [CRREL Translation: V. A. Kudryavtsev et al., *Fundamentals of Frost Forecasting in Geological Engineering Investigations* , CRREL Draft Translation 606, 1977, 489 pp.]

Menzel, L. (1997b): Modelling canopy resistances and transpiration of grassland. *Physics and Chemistry of the Earth, special issue: Atmospheric and Hydrological Processes and Models at the Soil Vegetation Atmosphere Interface.*

Kane, D. L., K. M. Hinkel, D. J. Goering, L. D. Hinzman, and S. I. Outcalt, 2001: Non-con ductive heat transfer associated with frozen soils, Global and Planetary Change 29, 275–292.

Kopp, M. (2017): A canopy energy balance model for the water balance model WaSiM. *Technical Report, Chair of Hydrology and River Basin Management, Department of Civil, Geo and Environmental Engineering, Technical University of Munich,* 11 pp.

Monteith, J.L. (1975): Vegetation and the atmosphere, vol. 1: Principles. *Academic Press, London*.

Muster, H. (1994): Ermittlung von Gebietsniederschlägen. *Beitrag im 18. Fortbildungslehrgang Hydrologie des DVWK „Niederschlag-Abfluss-Modelle für kleine Einzugsgebiete und ihre Anwendung",* 19. bis 23. September 1994, Karlsruhe, 31 S.

Nagare, R. M., Schincariol, R. A., Quinton, W. L., and M. Hayashi (2011): E ff ects of freezing on soil temperature, frost propagation and moisture redistribution in peat: laboratory investigations. *Hydrol. Earth Syst. Sci. Discuss., 8, 5387–5426, 2011* www.hydrol-earth-syst-sci-discuss.net/8/5387/2011/ doi:10.5194/hessd-8-5387-2011

Oke, T.R. (1987): Boundary Layer Climates, 2nd ed., *Routledge, London and New York*. 435 S.

Pack, R.T., D.G. Tarboton, C.N. Goodwin (1998?): SINMAP – A stability index approach to terrain stability hazard mapping. Terratech Consulting Ltd., Utah State University, C.N. Goodwin Fluvial System Consulting; User Manual

Penman, H.L. (1948): Natural evaporation from open water, bar soils and grass. *Proc. Royal Meteorologic Society, London*, 193 (A), 120-145.

Peschke, G. (1977): Ein zweistufiges Modell der Infiltration von Regen in geschichtete Böden. *Acta hydrophysica*, 22 (1), 39 - 48.

Peschke, G. (1987). Soil Moisture and Runoff Components from a Physically Founded Approach. *Acta hydrophysica*, 31 (3/4), 191-205.

Sarvas, R. (1974): Investigations on the annual cycle of development of forest trees II. Autumn dormancy and winter dormancy. *Communicationes Instituti Forestalis Fenniae*, 84 (1), 1-101.

Schulla, J. (1997): Hydrologische Modellierungvon Flussgebieten zur Abschätzung der Folgen von Klimaänderungen. Diss ETH 12018, Verlag Geographisches Institut ETH Zürich, 187 S.

Spaans, E. J. A. and Baker, J. M. (1996): The soil freezing characteristics: Its Measurement and Similarity to the Soil Moisture Characteristic. *Soil Sciences Soc. 60:13-19*

Stahl, K., R. D. Moore, J. M. Shea, D. Hutchinson, and A. J. Cannon (2008), Coupled modelling of glacier and streamflow response to future climate scenarios, Water Resour. Res. , 44 , W02422, doi:10.1029/2007WR005956

Tarboton, D.G., Bras, R.L. and Rodríguez-Iturbe, I. (1993): On the Extraction of Channel Networks from Digital Elevation Data. *In: Beven, K.J. and Moore, I.D. (ed.): Terrain Analysis and Distributed Modelling in Hydrology*. Orig. publ. in: *Hydrological Processes* 5(1), 1-126 (1991) und 6(3), 253-368 (1992).

Tetens, O. (1930): Über einige meteorologische Begriffe. *Zeitschrift für Geophysik, Braunschweig,* 6, 297-309.

Van Genuchten, M.T. (1976): A Closed-Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. *Soil Siences Society of American Journal*, Vol. 44(5), 892-898.

Wendling, U. (1975): Zur Messung und Schätzung der potentiellen Verdunstung. *Zeitschrift für Meteorologie*, 25 (2), 103-111.

Warscher, M., U. Strasser, G. Kraller, T. Marke, H. Franz, and H. Kunstmann (2013), Performance of complex snow cover descriptions in a distributed hydrological model system: A case study for the high Alpine terrain of the Berchtesgaden Alps, Water Resour. Res., 49, 2619–2637, doi:10.1002/wrcr.20219.

Williams, P. J. (1964): Unfrozen water content of frozen soils and soil moisture suction. Geotechnique, Vol. 14, No. 3, Sept. 1964, p. 231-246.

Winstral, A., and D. Marks (2002), Simulating wind fields and snow redistribution using terrain-based parameters to model snow accumulation and melt over a semi-arid mountain catchment, Hydrol. Processes, 16, 3585–3603.

Zhang, Y., S. K. Carey, W. L. Quinton, J. R. Janowicz, J. W. Pomeroy, and G. N. Flerhinger , 2010: Comparison of algorithms and parameterisations for infiltration into organic-covered permafrost soils, Hydrol. Earth Syst. Sci., 14, 729–750, www.hydrol-earth-syst-sci.net/14/729/2010/doi:10.5194/hess-14-729-2010

7 Appendices

7.1 Example of a control file for WaSiM

The sample control file contains all model parameters, file names and control parameters which are used for the modeling of the Thur-Basin. The file is well commented and should be looked at by using also the other chapters of this documentation (each section contains specific control file examples). If WaSiM is started, the control file is read in. All symbols are replaced, all comments are removed and the resulting file is written to the disk with the extension changed to ".$$$". Also this file should be consulted if unknown problems occur. Sometimes the reason for problems is a misuse of comments in the control file leaving not allowed lines in the control file. Only the translated control file is read in by each submodel to get the required names, model and control parameters. The rows of the control file are not allowed to be longer than 2000 characters, otherwise WaSiM may read undefined values (exception: land use and soil table entries and routing descriptions may be as long as required – they will be read in until a new record starts as a single line, even if they are separated by line breaks in the control file).

The following control file is not complete. Several sections are only partly copied into this documentation, since the entire file would be too long. Also, the comments following almost every entry are cut at column 100, sometimes the comments are completely removed. This control file example should give an overview of how the sections are ordered and how the variables are used etc. A real reference would be a working control file only.

```
# == protected part to be changed by the module adapter ==
# == all manual changes in this section will be overwritten by the module adapter ==

$set $mainpath       = d:\data\mybasin\
$set $InitialStateDirectory = $mainpath//StateIni\
$set $DefaultOutputDirectory = $mainpath//output\
$set $inpath_grid    = $mainpath//input\grids\
$set $inpath_meteo   = $mainpath//input\meteo\
$set $inpath_hydro   = $mainpath//input\hydro\
$set $inpath_ini     = $mainpath//input\ini\

$set $time           = 60

$set $starthour      = 1
$set $startday       = 3
$set $startmonth     = 1
$set $startyear      = 2000
$set $endhour        = 24
$set $endday         = 15
$set $endmonth       = 1
$set $endyear        = 2000

# it is important to set $outpath to an empty string in order to activate $DefaultOutputDirectory
$set $outpath        =

# readgrids : 1 = read storage grids (as SI, SSNOW,SLIQ...) from hard disk, 0=generate and
initialize
$set $readgrids     =  1

# read grids for dynamic phenology -> usually chilling grid should be read in if availabe because
othe
$set $DPreadgrids   =  1

# == end of protected part ==
$set $time           = 60.0
$set $year           = 96

# it is important to set $outpath to an empty string in order to activate $DefaultOutputDirectory
$set $outpath        =

# variables for parameters in unsatzon model: since several subbasins will be parameterized with
ident
```

```
# it is very convenient to have them here defined as variables, so we have only 18 parameter sets
inst

$set $grid           =  r500
$set $stack          =  r500
$set $suffix         =  grd
$set $code           =  s

# variables for standardgrids
# first section: grids, which differ for different subdivisions of the basin
$set $zone_grid             =  $grid//.ezg
$set $subcatchments         =  $grid//.ezg
$set $flow_time_grid        =  $grid//.fzs
$set $river_links_grid      =  $grid//.lnk
$set $regio_grid            =  $grid//.reg

#second section: grids, which doesn't depend on subdivision (only pixel-values are of interest)
$set $elevation_model       =  $grid//.dhm
$set $RelCellArea_grid      =  $grid//.rca
$set $CellSizeX_grid        =  $grid//.csx
$set $CellSizeY_grid        =  $grid//.csy
$set $slope_grid            =  $grid//.slp
$set $FlowDirection_grid    =  $grid//.fld
$set $aspect_grid           =  $grid//.exp
$set $land_use_grid         =  $grid//.use
$set $ice_firn_grid         =  $grid//.glc
$set $field_capacity_grid   =  $grid//.nfk
$set $ATBgrid               =  $grid//.atb
$set $hydr_cond_grid        =  $grid//.k
$set $soil_types            =  $grid//.soil
$set $sky_view_factor_grid  =  $grid//.hor
$set $drain_depth_grid      =  $grid//.drn
$set $drain_distance_grid   =  $grid//.dis
$set $irrigationcodes       =  $grid//.irr
$set $max_pond_grid         =  $grid//.maxpond
$set $clay_depth_grid       =  $grid//.cly
$set $river_depth_grid      =  $grid//.dep1
$set $river_width_grid      =  $grid//.wit
$set $tracer_1              =  $grid//.c1
$set $tracer_2              =  $grid//.c2
$set $tracer_3              =  $grid//.c3
$set $tracer_4              =  $grid//.c4
$set $tracer_5              =  $grid//.c5
$set $tracer_6              =  $grid//.c6
$set $tracer_7              =  $grid//.c7
$set $tracer_8              =  $grid//.c8
$set $tracer_9              =  $grid//.c9
$set $kolmationsgrid        =  $grid//.kol5
$set $gw_kx_1_grid          =  $grid//.kx1_025
$set $gw_kx_2_grid          =  $grid//.kx2
$set $gw_kx_3_grid          =  $grid//.kx3
$set $gw_ky_1_grid          =  $grid//.ky1_025
$set $gw_ky_2_grid          =  $grid//.ky2
$set $gw_ky_3_grid          =  $grid//.ky3
$set $gw_bound_h_1_grid     =  $grid//.bh1
$set $gw_bound_h_2_grid     =  $grid//.bh2
$set $gw_bound_h_3_grid     =  $grid//.bh3
$set $gw_bound_q_1_grid     =  $grid//.bq1
$set $gw_bound_q_2_grid     =  $grid//.bq2
$set $gw_bound_q_3_grid     =  $grid//.bq3
$set $aquiferthick1         =  $grid//.aq1d
$set $aquiferthick2         =  $grid//.aq2
$set $aquiferthick3         =  $grid//.aq3
$set $gw_storage_coeff_1    =  $grid//.s01
$set $gw_storage_coeff_2    =  $grid//.s02
$set $gw_storage_coeff_3    =  $grid//.s03
$set $gw_kolmation_1        =  $grid//.gk1
$set $gw_kolmation_2        =  $grid//.gk2
$set $gw_kolmation_3        =  $grid//.gk3
$set $lake_grid             =  $grid//.lak
$set $taucrit_grid          =  $grid//.tau
$set $ThawCoeffPermaFrost   =  $grid//.alpha
$set $debris_on_glaciers    =  $grid//.debris

# grids for surface hydrology modules
$set $forcingunitsgrid1    =  forc1//$grid//.//$suffix
$set $TStartPhenoGrid1     =  phen1//$grid//.//$suffix
$set $chillingunitsgrid1   =  chill1//$grid//.//$suffix
$set $FStargrid1           =  fstar1//$grid//.//$suffix
$set $forcingunitsgrid2    =  forc2//$grid//.//$suffix
```

```
$set $TStartPhenoGrid2     =  phen2//$grid//.//$suffix
$set $chillingunitsgrid2   =  chill2//$grid//.//$suffix
$set $FStargrid2           =  fstar2//$grid//.//$suffix
$set $forcingunitsgrid3    =  forc3//$grid//.//$suffix
$set $TStartPhenoGrid3     =  phen3//$grid//.//$suffix
$set $chillingunitsgrid3   =  chill3//$grid//.//$suffix
$set $FStargrid3           =  fstar3//$grid//.//$suffix
$set $albedo               =  albe//$grid//.//$suffix
$set $soilstoragegrid      =  sb__//$grid//.//$suffix
$set $throughfall          =  qi__//$grid//.//$suffix
$set $snowcover_outflow    =  qsno//$grid//.//$suffix
$set $melt_from_snowcover  =  qsme//$grid//.//$suffix
$set $days_snow            =  sday//$grid//.//$suffix
$set $snow_age             =  sage//$grid//.//$suffix
$set $snow_rate            =  snow//$grid//.//$suffix
$set $rain_rate            =  rain//$grid//.//$suffix
$set $firn_melt            =  qfir//$grid//.//$suffix
$set $ice_melt             =  qice//$grid//.//$suffix
$set $preci_grid           =  prec//$grid//.//$suffix
$set $preci_grid1          =  prec1//$grid//.//$suffix
$set $preci_grid2          =  prec2//$grid//.//$suffix
$set $irrig_grid           =  irri//$grid//.//$suffix
$set $etr2etpgrid          =  er2ep//$grid//.//$suffix
$set $tempegrid            =  temp//$grid//.//$suffix
$set $tempegrid1           =  temp1//$grid//.//$suffix
$set $tempegrid2           =  temp2//$grid//.//$suffix
$set $windgrid             =  wind//$grid//.//$suffix
$set $sunshinegrid         =  ssd_//$grid//.//$suffix
$set $radiationgrid        =  rad_//$grid//.//$suffix
$set $humiditygrid         =  humi//$grid//.//$suffix
$set $vaporgrid            =  vapo//$grid//.//$suffix
$set $ETPgrid              =  etp_//$grid//.//$suffix
$set $EIPgrid              =  eip_//$grid//.//$suffix
$set $ETRgrid              =  etr_//$grid//.//$suffix
$set $EVAPgrid             =  evap//$grid//.//$suffix
$set $EVARgrid             =  evar//$grid//.//$suffix
$set $ETRSgrid             =  etrs//$grid//.//$suffix
$set $SSNOgrid             =  ssno//$grid//.//$suffix
$set $SLIQgrid             =  sliq//$grid//.//$suffix
$set $SSTOgrid             =  ssto//$grid//.//$suffix
$set $sat_def_grid         =  sd__//$grid//.//$suffix
$set $SUZgrid              =  suz_//$grid//.//$suffix
$set $SIFgrid              =  sif_//$grid//.//$suffix
$set $EIgrid               =  ei__//$grid//.//$suffix
$set $SIgrid               =  si__//$grid//.//$suffix
$set $ExpoCorrgrid         =  exco//$grid//.//$suffix
$set $Tcorrgrid            =  tcor//$grid//.//$suffix
$set $Shapegrid            =  shap//$grid//.//$suffix
$set $INFEXgrid            =  infx//$grid//.//$suffix
$set $SATTgrid             =  satt//$grid//.//$suffix
$set $Nagrid               =  na__//$grid//.//$suffix
$set $SSPgrid              =  ssp_//$grid//.//$suffix
$set $Peakgrid             =  peak//$grid//.//$suffix
$set $SBiagrid             =  sbia//$grid//.//$suffix
$set $fcia_grid            =  nfki//$grid//.//$suffix
$set $tavg_grid            =  tavg//$grid//.//$suffix

# now variables for unsaturated zone model
$set $SB_1_grid      =  sb05//$grid//.//$suffix
$set $SB_2_grid      =  sb1_//$grid//.//$suffix
$set $ROOTgrid       =  wurz//$grid//.//$suffix
$set $QDgrid         =  qd__//$grid//.//$suffix
$set $QIgrid         =  qifl//$grid//.//$suffix
$set $GWdepthgrid    =  gwst//$grid//.//$suffix
$set $GWthetagrid    =  gwth//$grid//.//$suffix
$set $GWNgrid        =  gwn_//$grid//.//$suffix
$set $UPRISEgrid     =  uprs//$grid//.//$suffix
$set $PERCOLgrid     =  perc//$grid//.//$suffix
$set $GWLEVELgrid    =  gwlv//$grid//.//$suffix
$set $QDRAINgrid     =  qdrn//$grid//.//$suffix
$set $QBgrid         =  qb__//$grid//.//$suffix
$set $GWINgrid       =  gwin//$grid//.//$suffix
$set $GWEXgrid       =  gwex//$grid//.//$suffix
$set $act_pond_grid  =  pond//$grid//.//$suffix
$set $MACROINFgrid   =  macr//$grid//.//$suffix
$set $SUBSTEPSgrid   =  step//$grid//.//$suffix

$set $SnowFreeDaysGrid = sfre//$grid//.//$suffix
$set $SnowCoverDaysGrid = scov//$grid//.//$suffix
$set $ThawDepthGrid    = thdp//$grid//.//$suffix
```

```
# variables for groundwater modeling
$set $flowx1grid     =  gwx1//$grid//.//$suffix
$set $flowx2grid     =  gwx2//$grid//.//$suffix
$set $flowx3grid     =  gwx3//$grid//.//$suffix
$set $flowy1grid     =  gwy1//$grid//.//$suffix
$set $flowy2grid     =  gwy2//$grid//.//$suffix
$set $flowy3grid     =  gwy3//$grid//.//$suffix
$set $head1grid      =  gwh1//$grid//.//$suffix
$set $head2grid      =  gwh2//$grid//.//$suffix
$set $head3grid      =  gwh3//$grid//.//$suffix
$set $GWbalance1grid =  gwbalance1//$grid//.//$suffix
$set $GWbalance2grid =  gwbalance2//$grid//.//$suffix
$set $GWbalance3grid =  gwbalance3//$grid//.//$suffix

# result grids for surface routing model
$set $surfspeed_grid        =  sfcv//$grid//.//$suffix
$set $surfflux_grid         =  sflx//$grid//.//$suffix

# some new stacks and grids for the dynamic glacier model
$set $firn_WE_stack           = glfirn//$stack//.//$suffix
$set $GlacierMassBalance      = glmb//grid//.//$suffix
$set $OldGlacierMassBalance   = glmb_old//grid//.//$suffix
$set $glacierizedCells_grid   = glc_//$grid//.//$suffix
$set $glacier_codes_grid      = glid//$grid//.//$suffix

# result-stacks for Unsatzonmodel
$set $Thetastack              = teth//$stack//.//$suffix
$set $hydraulic_heads_stack   = hhyd//$stack//.//$suffix
$set $geodetic_altitude_stack = hgeo//$stack//.//$suffix
$set $flowstack               = qu__//$stack//.//$suffix
$set $concstack               = conc//$stack//.//$suffix


# parameters for interpolation of meteorological input data
$set $SzenUse        =  0
$set $IDWmaxdist     =  200000
$set $IDWweight      =  2
$set $Anisoslope     =  0.0
$set $Anisotropie    =  1.0


# explanation of writegrid and outputcode some lines below
$set $Writegrid      =  3
$set $Writestack     =  3

$set $once_per_interval  =  2001
$set $avrg_per_24Invs    =  2024
$set $sum_per_24Invs     =  4024
$set $routing_code       =  5001


#  Writegrid : max. 4 digits (nnnn)
#
# only if writegrid >= 1000: 1. digit (1nnn, or 2nnn)
# 0 = no vegetation period based grid is written
# 1 = sum grid is written for vegetation period (summing up each value as long as this cells veget
# 2 = average value grid is written for vegetation period (summing up each value as long as this c
# only if writegrid >= 100: 2. digit (n1nn, or n2nn or n3nn or 1nn..3nn  -> leading digits may be
#  0 = no minimum or maximum grid is written
#  1 = minimum grid is written (minimum value for each of the grid cells over the entire model per
#  2 = maximum grid is written (maximum value for each of the grid cells over the entire model per
#  3 = both grids are written (minimum and maximum value for each of the grid cells over the entir
# only if Writegrid >=  10: 3rd digit: sums or means (1n ... 8n or n1n..n8n or nn1n..nn8n  -> lead
#   0 = no sum grid will be written
#   1 = one sum grid will be written at the end of the model run
#   2 = one sum grid per model year
#   3 = one sum grid per model month
#   4 = one sum grid per day (only, if timestep < 1 day)
#   5 = one mean value grid at the end of the model run
#   6 = one mean value grid per model year
#   7 = one mean value grid per month
#   8 = one mean value grid per day
# last digit (nnn1 .. nnn5 or nn1..nn5 or n1..n5 or 1..5 -> leading digits may be omitted) (for ac
#    1 = (over)write each timestep into the same grid (for security in case of model crashs)
#    2 = write grids each timestep to new files, the name is build from the first 4 letters
#        of the regular grid name and then from the number of month, day and hour (hoer as file ex
#        example: tempm500.grd will become prec0114.07 for 14.January, 7:00.
#    3 = only the last grid of the model run will be stored
#    4 = the grid from the last hour of each day (24:00) will be stored (for each day the same fil
```

```
#    5 = like 4, but each day a new grid file is created (like for code 2)
#    6 = actual grid at the end of each month
#    7 = actual grid at the end of each year
#    8 = write immediately after reading the grid from file and filling missing values. This is us
#
# outputcode (for statistic files for zones or subcatchments)
#
# the Codes behind the names of the statistic files have the meaning of:
# <1000 no output
# 1<nnn> spatial mean values for the entire basin, averaged in time over <nnn> intervals (timesteps
# 2<nnn> spatial mean values for all zones (subbasin) and for the entire basin, averaged in time ov
# 3<nnn> spatial means for the entire basin, added up in time over <nnn> intervals (timesteps)
# 4<nnn> spatial means for all zones (subbasin) and for the entire basin, added up in time over <nn
# 5<nnn> spatial means for the entire basin and for those subbasins which are specified in the outp
# 6<nnn> spatial means for the entire basin and for those subbasins which are specified in the outp
#
# example:
# 2001 = per timestep for all subcatchments (and entire basin) one (spatially averaged) value
# 2004 = every 4 steps one averaged value over the last 4 time steps for all subcatchments and fo
# 4024 = Sums of the mean subbasins/entire basin values of the timesteps over 24 timesteps (e.g. d
# 3120 = averaged values (over 120 time steps!) only for the entire basin (spatially averaged)
# 5012 = averaged values (over 12 timesteps) as spatial averages for the entire basin and for each
of

[output_list]
17               # number of subbasins which are scheduled for output (is only of interest, if the
code
1
2
3
4
5
6
7
8
9
10
11
12
13
15
21
22
23
99

[output_interval]
25              # increment of time steps until an output to the screen is done (24 = each day one
outp
1 # warning level for interpolation (no station within search radius)
0 # unit of routed discharge (0=mm/timestep, 1=m3/s)
0 # minutes from the hour-entry in the input data files until the end
# of the time step is reached 0 if the end of time step is given like "84 01 01 01",
# but it should be $time if the begin is given like in "84 01 01 00"
WriteAsciiGrids = 1 # 0 if grids should be written in WaSiM native
InitialStateDirectory = $InitialStateDirectory # if using this parameter, all state grids  as
DefaultOutputDirectory = $DefaultOutputDirectory # this is the default output directory, all ou
# there are some exceptions, though: for external coupling no default output path is used
# relative pathnames may be used as well.

[coordinates]
46.1 # geogr. latitude (center of the basin -> for radiation calculations)
7.2  # geogr. longitude (center of the basin)
15.0 # meridian according to the official time (middle europe: 15)(east: 0 ... +180 degree,
1    # time shift of Meteo-data-time with respect to the true local time (mean sun time)
# e.g.: if meteo-data are stored in UTC-time and the time meridian is 15 east (central europe),
# the local time is 1 hour later than the time in the meteo-data-file, so 1 hour has to be added
# this is important for calculation of sunshine duration and radiation

[region_transition_distance]
10000 # in m

[soil_surface_groundwater_substeps].
1 # number of sub time steps for module group surface routing, unsaturated zone model and groundwa
# Values to start with are 1 (default), 2
# (half of the common time step), 3 etc. be careful to set too high values here since the mode
# performance will go down dramatically, since unsatzonmodel and surface routing are called each
# time!

[elevation_model]
```

```
$inpath_grid//$elevation_model    # grid with the digital elevation data

[zone_grid]
$inpath_grid//$zone_grid          # grid with Zone codes

$set $lai_grid = lai_//$grid//.//$suffix
$set $z0_grid = z0_//$grid//.//$suffix
$set $root_grid = root_//$grid//.//$suffix
$set $rse_grid = rse_//$grid//.//$suffix
$set $rsi_grid = rsi_//$grid//.//$suffix
$set $rsc_grid = rsc_//$grid//.//$suffix
$set $albedo_grid = albedo_//$grid//.//$suffix
$set $vcf_grid = vcf_//$grid//.//$suffix

$set $lai_stat = lai_//$grid//.//$code//$year
$set $z0_stat = z0_//$grid//.//$code//$year
$set $root_stat = root_//$grid//.//$code//$year
$set $rse_stat = rse_//$grid//.//$code//$year
$set $rsi_stat = rsi_//$grid//.//$code//$year
$set $rsc_stat = rsc_//$grid//.//$code//$year
$set $albedo_stat = albedo_//$grid//.//$code//$year
$set $vcf_stat = vcf_//$grid//.//$code//$year


# there is a simple possibility starting with WaSiM 8.10.03 to do the nearest neighbor filling
permane
# will be writen to the default output directory with it's original name but an additional suffix
"fil
#used as input grid (without fillcode = 1 then).

[standard_grids]
28               # number of standard grids
# path           # identification           # fillcode 0=no, 1=yes (fill missing v
$inpath_grid//$slope_grid  slope_angle  1 # grid with slope angl
$inpath_grid//$aspect_grid slope_aspect 1 # grid with slope aspe
$inpath_grid//$regio_grid  regression_regions fillcode = 1 # writecode = 8 readcod
$inpath_ini//$lai_grid     leaf_area_index1   fillcode = 1 defaultValue = 3       writec
$inpath_ini//$z0_grid      RoughnessLength1 fillcode = 1 defaultValue = 0.1 writec
$inpath_ini//$root_grid    root_depth1         fillcode = 1 defaultValue = 1.0 writec
$inpath_ini//$vcf_grid     vegetation_coverage_degree1      fillcode = 1 defaultValue = 0.9 writec
$inpath_ini//$rse_grid     SurfaceEvaporationResistance     fillcode = 1 defaultValue = 300 writec
$inpath_ini//$rsi_grid     SurfaceIntercepResistance1       fillcode = 1 defaultValue = 5   writec
$inpath_ini//$rsc_grid     SurfaceCanopyResistance1         fillcode = 1 defaultValue = 75  writec
$inpath_ini//$albedo_grid  albedo  fillcode = 1 defaultValue = 0.2 writec
etc.. reading all standard grids

# variable grids are used by more than one module or can be changed (like albedo and soil storage)
$set $SurfStorSiltingUp    =  sfstsu//$grid//.//$suffix
$set $pondgridtopmodel     =  pond_top//$grid//.//$suffix
$set $VegetationStart      =  vegstart//$grid//.//$suffix
$set $VegetationStop       =  vegstop//$grid//.//$suffix
$set $VegetationDuration   =  vegduration//$grid//.//$suffix

[variable_grids]
2                                                          # Number of variable grids to read
$outpath//$glacierizedCells_grid GlacierizedCells 0  -9999  # glacierized fraction of each cell (0
7 #$Writegrid                                                      # Writegrid for glacerized cells
1 #$readgrids                                              # should always be 1 since otherwise n
$outpath//$glacier_codes_grid    GlacierCodes     0  -9999  # codes for each single glacier. This
7 #$Writegrid                                                      # Writegrid for glacier codes
1 #$readgrids                                              # should always be 1 since otherwise n


[model_time]
$starthour            #  start hour
$startday             #  start day
$startmonth           #  start month
$startyear            #  start year
$endhour              #  end hour
$endday               #  end day
$endmonth             #  end month
$endyear              #  end year

[meteo_data_count]
8

[meteo_names]
temperature_reg1
temperature_reg2
precipitation_reg1
```

```
precipitation_reg2
wind_speed
air_humidity
vapor_pressure
global_radiation
sunshine_duration


# methods:
# 1  = idw
# 2  = regress
# 3  = idw+regress
# 4  = thiessen
# 5  = bilinear
# 6  = bilinear gradients and residuals linarly combined,
# 7  = bicubic spline,
# 8  = bicubic splines of gradients and residuals linearly combined,
# 9  = read grids according to the name in a grid list file,
# 10 = regression from Stationdata instead from outputfile of regr.exe (similar to method 1, except
th
# 11 = regression and IDW from station data (equivalent to method 3, except that no station
selection
# 12 = Thiessen with given lapse rate (as single next line parameter or with multiple parameters
lower


[temperature_reg1] # see section on interpolation for description of parameters
10
$inpath_meteo//t2m_reg1_9610.dat AdditionalColumns=0
#$inpath_meteo//t2m_reg1.out
$outpath//$tempegrid1
5//$Writegrid
1.0
$outpath//t2m_reg1_//$grid//.//$code//$year $sum_per_24Invs
998
$IDWweight
0.2
$IDWmaxdist
-65
0.8
-40
-40
40
40
$SzenUse
1
4

etc.. describe all meteorologic variables here as listed in [meteo_names] and [meteo_data_count]

# ----------  parameter for model components  -----------------

[RegionalSuperposition]
1
$time
NumberOfEntities = 2;
temperature {
       entityinputgrid = temperature_reg1 ;
              regions = 1   2   ;
              weights = 1.0 0.0 ;
       entityinputgrid = temperature_reg2 ;
              regions = 1   2   ;
              weights = 0.0 1.0 ;
       outputgrid  = $outpath//$tempegrid ;
              writecode = 5//$Writegrid ;
       outputtable = $outpath//t2m_//$grid//.//$code//$year;
              statcode  = $once_per_interval;
}
precipitation {
       entityinputgrid = precipitation_reg1 ;
              regions = 1   2   ;
              weights = 1.0 0.0 ;
       entityinputgrid = precipitation_reg2 ;
              regions = 1   2   ;
              weights = 0.0 1.0 ;
       outputgrid  = $outpath//$preci_grid ;
              writecode = 1//$Writegrid ;
       outputtable = $outpath//prec//$grid//.//$code//$year;
              statcode  = $once_per_interval;
}
```

```
[precipitation_correction]
1                     # 0=ignore this module, 1 = run the module
0.0                   # Snow-rain-temperature
1.02                  # liquid:   b in:  y = p(ax + b)
0.01                  # liquid:   a in:  y = p(ax + b) = 1% more per m/s + 0.5% constant
1.15                  # Snow:     b in:  y = p(ax + b)
0.15                  # Snow:     a in:  y = p(ax + b) = 15% more per m/s + 45% constant

# correction factors for direct radiation are calculated
# if the cell is in the shadow of another cell, or if a cell is not in the sun (slope angle!)
# then the factor is 0.
# control_parameter: 1 = radiation correction WITH  shadow WITHOUT temperature correction
#                    2 = radiation correction WITH  shadow WITH  temperature correction
#                    3 = radiation correction WITHOUT shadow WITHOUT temperature correction,
#                    4 = radiation correction WITHOUT shadow WITH  Temperatur

[radiation_correction]
1                                                       #  0=ignore  this  module,  1  =  run  the
module
$time                                                   # duration of a time step in minutes
2                                                       #   control   parameter   for   radiation
correction (
$outpath//$Tcorrgrid                                 # name of the grids with the corrected tempera
5//$Writegrid                                        # Writegrid for corrected temperatures
20                                                   # scaling factor for temperature correction
$outpath//$ExpoCorrgrid                              # name of the grids with the correction factor
5//$Writegrid                                        # Writegrid
$outpath//$Shapegrid                                 # name of the grids for codes 1 for theor. sha
5//$Writegrid                                        # Writegrid
1                                                       # interval counter, after reaching this
value,
1                                                       # Spitting of the interval, usefull for
time s


[evapotranspiration]
1                                                    # 0=ignore this module, 1 = run the module
$time                                                # duration of a time step in minutes
1                                                    # Method: 1=Penman-Monteith, 2=Hamon (only dai
0.5 0.6 0.8 1.0 1.1 1.1 1.2 1.1 1.0 0.9 0.7 0.5     # PEC correction factor for HAMON-evapotranspi
0.20 0.20 0.21 0.29 0.29 0.28 0.26 0.25 0.22 0.22 0.20 0.20 # fh (method 4: Haude) monthly factors
0.5                                                  # fk -> factor for Wendling-evapotranspiration
$outpath//$ETPgrid                                   # result grid for pot. evapotranspiration in m
$Writegrid            # 0, if no grid-output is needed, else one of the code
$outpath//etp_//$grid//.//$code//$year $sum_per_24Invs
$outpath//$ETRgrid
$Writegrid
$outpath//etr_//$grid//.//$code//$year $sum_per_24Invs
$outpath//$EVAPgrid
$Writegrid
$outpath//evap//$grid//.//$code//$year $sum_per_24Invs
$outpath//$EVARgrid
$Writegrid
$outpath//evar//$grid//.//$code//$year $sum_per_24Invs
$outpath//$ETRSgrid
$Writegrid
$outpath//etrs//$grid//.//$code//$year $sum_per_24Invs
$outpath//$EIPgrid
$Writegrid
$outpath//eip_//$grid//.//$code//$year $sum_per_24Invs
$outpath//rgex//$grid//.//$code//$year $sum_per_24Invs
+0.23   +1.77    -2.28    +1.28
+0.072  -0.808   +2.112   -0.239
0.88 0.05
1654.0
3.3 4.4 6.1 7.9 9.4 10.0 9.9 9.0 7.8 6.0 4.2 3.2
0.62  0.1

[snow_model]
1                     # 0=ignore this module, 1 = run the module
$time                 # duration of a time step in minutes
1                     # method 1=T-index, 2=t-u-index, 3=Anderson comb., 4=extended com.
1.0                   # transient zone for rain-snow (T0R +- this range)
0.0                   # T0R    temperature limit for rain (Grad Celsius)
0.0                   # T0     temperature limit snow melt
0.05                   # CWH    storage capacity of the snow for water (relative part)
1.0                   # CRFR   coefficient for refreezing
1.8                   # C0     degree-day-factor mm/d/C
```

```
0.8                 # C1     degree-day-factor without wind consideration  mm/(d*C)
0.17                # C2     degree-day-factor considering wind mm/(d*C*m/s)
0.07                # z0     roughness length cm for energy bilance methods (not used)
1.0                 # RMFMIN minimum radiation melt factor      mm/d/C comb. method
2.5                 # RMFMAX maximum radiation melt factor      mm/d/C comb. method
0.45                # Albedo for snow (Min)
0.90                # Albedo for snow (Max)
$outpath//$rain_rate                                # rain rate
$Writegrid                                          # 0, if no grid-output is needed, else one of
the code
$outpath//rain//$grid//.//$code//$year $sum_per_24Invs      # rain rate
$outpath//$snow_rate                                # snow rate
$Writegrid                                          # 0, if no grid-output is needed, else one of
the code
$outpath//snow//$grid//.//$code//$year $sum_per_24Invs      # snow rate
$outpath//$days_snow                                # days with snow (SWE > 5mm)
$Writegrid                                          # 0, if no grid-output is needed, else one of
$outpath//sday//$grid//.//$code//$year $sum_per_24Invs      # days with snow (SWE > 5mm)
$outpath//$snow_age                                 # snow age (days without new snow)
$Writegrid                                          # 0, if no grid-output is needed, else one of
$outpath//sage//$grid//.//$code//$year $sum_per_24Invs      # days since last snowfall
$outpath//albe//$grid//.//$code//$year $sum_per_24Invs      # Albedo
$outpath//$snowcover_outflow                # discharge from snow, input (precipitation) f
$Writegrid                                          # 0, if no grid-output is needed, else one of
$outpath//qsch//$grid//.//$code//$year $once_per_interval   # melt flow (or rain, if there is no s
$outpath//$melt_from_snowcover              # discharge from snow, input (precipitation) f
$Writegrid                                          # 0, if no grid-output is needed, else one of
the code
$outpath//qsme//$grid//.//$code//$year $once_per_interval   # melt flow in mm/dt
$outpath//$SSNOgrid                                 # name of the grids with the snow storage soli
$Writegrid                                          # 0, if no grid-output is needed, else one of
the code
$outpath//$SLIQgrid                                 # name of the grids with the snow storage liqu
$Writegrid                                          # 0, if no grid-output is needed, else one of
the code
$outpath//ssto//$grid//.//$code//$year $once_per_interval   # total snow storage, in mm, (liquid a
$outpath//$SSTOgrid                                 # name of the grids with the total snow storag
$Writegrid                                          # 0, if no grid-output is needed, else one of
the code
$readgrids                                          # 1=read snow storage solid, liquid grids from

[ice_firn]
12    # method for glacier melt: 1=classical t-index, 2=t-index with correction by radiation, 11 =
dyna
6.0                                         # t-index factor for ice
5.0                                         # t-index factor for firn
4.0                                         # t-index factor for snow
1.8                                                 # melt factor
-0.00010                                    # radiation coefficient for ice_min  (for meth
+0.00070                                    # radiation coefficient for ice_max  (for meth
+0.00006                                    # radiation coefficient for snow_min (for meth
+0.00030                                            # radiation coefficient for snow_max (for meth
12   # els-konstante for ice
120  # els-konstante for firn
24   # els-konstante for snow
0.01  # initial reservoir content for ice discharge (single linear storage approach)
0.01  # initial reservoir content for firn discharge (single linear storage approach)
0.01  # initial reservoir content for snow discharge (single linear storage approach)
$outpath//$firn_melt                                # melt from firn
$Writegrid                                          # 0, if no grid-output is needed, else one of
the code
$outpath//qfir//$grid//.//$code//$year $once_per_interval   # melt from firn as statistic file
$outpath//$ice_melt                                 # melt from ice
$Writegrid                                          # 0, if no grid-output is needed, else one of
the code
$outpath//qice//$grid//.//$code//$year $once_per_interval   # melt from ice as statistic file
$outpath//qglc//$grid//.//$code//$year $once_per_interval   # discharge from snow, ice and firn as
# ------------------------------------------------------------------------
# now some new parameters for the new dynamic glacier model (methods 11 and 12)
$outpath//qsgl//$grid//.//$code//$year $once_per_interval   # melt from snow from glacier only as
$readgrids                                  # 1=read grids and stacks from disk, 0=generat
7 2700 1.8                                  # number of layers for the firn stack, followe
09 30                                       # month and day (hour is set automatically to
28.5 1.36 10 4                              # VAscaling and VAexponent for Volume-Area-Rel
$outpath//$firn_WE_stack                    # water equivalent for firn (given as stack, n
$Writestack                                 # 0, if no grid-output is needed, else one of
$outpath//glfirn//$grid//.//$code//$year $once_per_interval
$outpath//$GlacierMassBalance               # output grid with mass balance of the glacier
$Writegrid                                  # 3: write at end of simulation (impor
$outpath//$OldGlacierMassBalance            # output grid with mass balance of the glacier
```

```
$Writegrid                                          # 3: write at end of simulation (impor
$outpath//glmb//$grid//.//$code//$year $sum_per_24Invs
$outpath//glmb2//$grid//.//$code//$year $sum_per_24Invs
1.0    # additional parameter when using a debris grid: global scaling factor for debris factors

[permafrost]
1                    # method: 1=simple Alpha*sqrt(snow-free-days) approach to estimate thawdepth
30                   # number of days with snow cover after which the soil is assumed ...
5                    #  maximum  snow  water  equivalent  for  the  interval  to  be  counted  as  snow
covered

# methods for interception model, supported by version 10.04.00 and higher
# 1 = traditional/legacy: using ETP (no separate EIP), first snow model, then interception
# 2 = traditional/legacy: using separate EIP, first snow model, then interception
# 11 = using ETP (no separate EIP), first interception model, then snow model
# 12 = using separate EIP, first interception model, then snow model
#  21  =  using  ETP  (no  separate  EIP),  first  interception  model,  then  snow  model,  +  Snow  canopy
interception SCN
# 22 = using separate EIP, first interception model, then snow model, + Snow canopy interception
SCN
#  31  =  using  ETP  (no  separate  EIP),  first  interception  model,  then  snow  model,  +  Snow  canopy
interception SCN + Canopy energy balance CEB
# 32 = using separate EIP, first interception model, then snow model, + Snow canopy interception
SCN + Canopy energy balance CEB
[interception_model]
1                                                       #  0=ignore  this  module,  1  =  run  the
module
$time                                                   # duration of a time step in minutes
1                                                  # method: see comments above this section
$outpath//$throughfall                                  #  result  grid  :   =  outflow  from  the
intercepti
$Writegrid                                              # 0, if no grid-output is needed, else
one of
$outpath//qi__//$grid//.//$code//$year $sum_per_24Invs
$outpath//$EIgrid                                       # Interzeption evaporation, grid
$Writegrid                                         # 0, if no grid-output is needed, else one of
$outpath//ei__//$grid//.//$code//$year $sum_per_24Invs      # zonal statistic
$outpath//$SIgrid                                       #  storage  content  of  the  interception
storage
$Writegrid                                         # 0, if no grid-output is needed, else one of
$outpath//si__//$grid//.//$code//$year $sum_per_24Invs
0.35                                                    # layer thickness of the waters on the
leaves
$readgrids                                              # 1=read grids from disk, else generate
intern
# new parameters for snow interception model (implemented by Kristian Förster
# method must be 21, 22, 31 or 32 (where 31 and 32 will additionally switch on canopy
# energy balance, see above
$outpath//SCNTEMP//$grid//.//$suffix            # Sub-canopy temperature, grid
$Writegrid #_every                        # 0, if no grid-output is needed, else one of the codes
described above
$outpath//SCNTEMP_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNRH//$grid//.//$suffix              # Sub-canopy relative humidity, grid
$Writegrid #_every                        # 0, if no grid-output is needed, else one of the codes
described above
$outpath//SCNRH//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNRAD//$grid//.//$suffix             # Sub-canopy shortwave radiation, grid
$Writegrid #_every                       # 0, if no grid-output is needed, else one of the codes
described above
$outpath//SCNRAD_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNWIND//$grid//.//$suffix            # Sub-canopy temperature, grid
$Writegrid #_every                        # 0, if no grid-output is needed, else one of the codes
described above
$outpath//SCNWIND_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNLOAD//$grid//.//$suffix            # Snow interception load, grid
$Writegrid #_every                       # 0, if no grid-output is needed, else one of the codes
described above
$outpath//SCNLOAD_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNAGE//$grid//.//$suffix             # Age of intercepted snow, grid
$Writegrid #_every                        # 0, if no grid-output is needed, else one of the codes
described above
$outpath//SCNAGE_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNALB//$grid//.//$suffix             # Albedo of intercepted snow, grid
$Writegrid #_every                        # 0, if no grid-output is needed, else one of the codes
described above
$outpath//SCNALB_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNSUBL//$grid//.//$suffix            # Sublimation of intercepted snow, grid
$Writegrid #_every                        # 0, if no grid-output is needed, else one of the codes
described above
$outpath//SCNSUBL_//$grid//.//$code//$year $hour_mean # zonal statistic
```

```
$outpath//SCNUNLOAD//$grid//.//$suffix          # unload (incl. melt) of intercepted snow, grid
$Writegrid #_every                        # 0, if no grid-output is needed, else one of the codes
described above
$outpath//SCNUNLOAD_//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//SCNLIQOUT//$grid//.//$suffix            # liquid water output of the snow canopy model,
grid
$Writegrid #_every                        # 0, if no grid-output is needed, else one of the codes
described above
$outpath//SCNLIQOUT_//$grid//.//$code//$year $hour_mean # zonal statistic
# new algorithms for interception canopy energy balance, implemented by Matthias Kopp,
# methods must be 31 or 32, this includes snow canopy interception
$outpath//CEB_CANENBAL//$grid//.//$suffix
$Writegrid #_every
$outpath//CEB_CANENBAL//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//CEB_CANTEMP//$grid//.//$suffix
$Writegrid #_every
$outpath//CEB_CANTEMP//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//CEB_CANTEMP_OLD//$grid//.//$suffix
$Writegrid #_every
$outpath//CEB_CANTEMP_OLD//$grid//.//$code//$year $hour_mean # zonal statistic
$outpath//CEB_DUMMY//$grid//.//$suffix
$Writegrid #_every
$outpath//CEB_DUMMY//$grid//.//$code//$year $hour_mean # zonal statistic
0.4         # minimum LAI value above which the snow canopy model will be started
0.16        # minimum Z0 value above which the snow canopy model will be started
1.52          # LAI scaling factor for the snow canopy model --> CALIBRATION!!
[infiltration_model]
0                             # 0=ignore this module, 1 = run the module
$time                         # duration of a time step in minutes
$outpath//$INFEXgrid          # grid with infiltration excess in mm (surface runoff)
$Writegrid                #  for surface discharge (fraction 1)
$outpath//infx//$grid//.//$code//$year $once_per_interval
$outpath//$SATTgrid           # grid with code 1=saturation at interval start, 0 =no saturation.
$Writegrid                    # Writegrid for saturation code grids
0.1                           # fraction of reinfitrating water (of the infiltration excess)

$set $SDISPgrid            =  sdis//$grid//.//$suffix
$set $RPAUSgrid            =  paus//$grid//.//$suffix
$set $EKIN_grid            =  ekin//$grid//.//$suffix
$set $TSBB_grid            =  tsbb//$grid//.//$suffix
$set $QDSU_grid            =  qdsu//$grid//.//$suffix

[SiltingUpModel]
0                             # 0=ignore this module, 1 = run the module
$time                         # duration of a time step in minutes
1                              # method: 1=traditional (default if this line
$outpath//sdis//$grid//.//$code//$year $once_per_interval
$outpath//qdsu//$grid//.//$code//$year $once_per_interval
$outpath//$SDISPgrid          # grid with actual silting up disposition
$Writegrid                    # writegrid for this grid
$outpath//$RPAUSgrid
$Writegrid                    # writegrid for this grid
$outpath//$EKIN_grid          # grid with actual kinetic energy of the event
$Writegrid                    # writegrid for this grid
$outpath//$TSBB_grid          # grid with actual time since last soil tillage
$Writegrid                    # writegrid for this grid
$outpath//$QDSU_grid      # grid with direct runoff from silting up model (will be used in unsatzon
$Writegrid                    # writegrid for this grid
1     2     3     4     5     6     7     8     9     10    11    12    13
1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
$readgrids                    # readgrid code 0 do not read, 1 = read grids
65.1          # for method 2:  parameter A in I_0 = A (initial infiltration capacity...
12.21         # for method 2:  parameter B in I_end = B*(dg^C)*(fd^D) (in method 0 defined as 12.2)
0.521         # for method 2:  parameter C in I_end = B*(dg^C)*(fd^D) (in method 0 defined as 0.52)
-0.641        # for method 2:  parameter D in I_end = B*(dg^C)*(fd^D) (in method 0...
0.0131        # for method 2:  parameter E in Cv = E*(fd^F)*(dg^G)*(t_cult^H) (in method ...
-1.031        # for method 2:  parameter F in Cv = E*(fd^F)*(dg^G)*(t_cult^H) (in method
0.71          # for method 2:  parameter G in Cv = E*(fd^F)*(dg^G)*(t_cult^H) (in method
-0.191        # for method 2:  parameter H in Cv = E*(fd^F)*(dg^G)*(t_cult^H) (in method
SiltingUpExpressions { # please read the short documentation on the expression parser below ...
       W = ((P>0.05)&(P<76.2))*(11.89+8.73*log10(Abs(P+0.001))) + (P>=76.2)*28.33;
       X = A;
       Y = B*K^C*(L*100)^D;
       C1 = (100*L)^F;
       C2 = K^G;
       C3 = (O+0.001)^H;
       Z = (O<=0) + (O>0)*(E*C1*C2*C3);
       V = ((X-Y)*exp(-Z*Q)+Y)*R/60;
}
```

```
# Short description of the expression parser and the expression list syntax for method 3
# -    Expressions can be defined following algebraic rules:
# -    Each line contains a single expression which must be closed with a semi colon.
# -    Each assignment (e.g. A = 15) results in creating or updating a value in ...
# -    A number of values is already defined by WaSiM (as interface from the calling module), ...
# The expression parser  is based on the source code of the expression parser used in ...
# written by Jos de Jong, 2007. It was adopted to the usage in WaSiM by simplifying ...
# extracting the variable list as an external class (to be handled by WaSiM) and some other ...
#     Operators (ascending precedence per line, no precedence within a line):
#        & | << >>                          (AND, OR, BITSHIFTLEFT, BITSHIFTRIGHT)
#        = <> < > <= >=              (EQUAL, UNEQUAL, SMALLER, LARGER, SMALLEREQ, LARGEREQ)
#        + -                         (PLUS, MINUS)
#        * / % ||                    (MULTIPLY, DIVIDE, MODULUS, XOR)
#        ^                           (POW)
#        !                           (FACTORIAL)
#    Functions (must be used with brackets):
#        Abs(arg), Exp(arg), Sign(arg), Sqrt(arg), Log(arg), Log10(arg)
#        Sin(arg), Cos(arg), Tan(arg), ASin(arg), ACos(arg), ATan(arg)
#        Factorial(arg)
#    Variables:
#        Pi, Euler (not only e, e is a predefined variable used by WaSiM to deliver a value to the
#        you can define your own variables, even with with more than one significant character
#      there is no distinction between upper and lower case in function names and variables.
#    Other:
#        Scientific notation supported
#
# ====> what values WaSiM defines forinput (can be used in any expression)
# A to J: values as used in soiltable with names SU_PAR01 to SU_PAR10
# K: grain size distribution Dg, internally calculated after
#      double FClay   = log004+log2;
#      double FSilt    = 0.3326 * (log2+log6_3) +  0.3348 *(log6_3+log20)  + 0.1704 * (log20+log36)
+
#      double FSand   = 0.1336 * (log63+log100) + 0.2005 *(log100+log200) + 0.3318 *(log200+log630)
+
#      double FStones1 = (log2000+log6300);
#      double FStones2 = (log6300+log20000);
#      double FStones3 = (log20000+log63000);
#      double FStones4 = (log63000+log200000);
#      double   dg   =   (FClay*dFractionClay   +   FSilt*dFractionSilt   +   FSand*dFractionSand   +
FStones1*dFract
#      with fractions of each grain size class taken from the soil table
# L:   fraction of sand
# M:   fraction of clay
# N:   fraction of silt
# O:   t_cult, time since last soil cultivation (in days)
# P:   rain intensity in mm/h, taken from precipitation input
# Q:   e_kin: accumulated cinetic energy: for all expressions resulting in W, X, Y or Z: result
value
# R:   internal time step in minutes
# ====> What WaSiM expects for output: (ranging from Z downwards, will be used by WaSiM when going
ahe
# Z:   silting up disposition SDISP
# Y:   end infiltration rate i_inf
# X:   start infiltration rate i0
# W:   actual cinetic energy
# V:   potential infiltration rate inf_pot, depending on energy, siting up disposition, inf_start
and
# order of expressions evanulated by WaSiM:
#      expressions returning W, X, Y and Z are independently of each other.
#      expression V must be called as last call in any case, since WaSiM will update EKIN
internally
#      other expressions for storing intermediate results may be defined at any position in the
expre

[SurfaceRoutingModel]
0                           # 0=ignore this module, 1 = run the module
$time                       # duration of a time step in minutes
2                        # method: 1=MultipleFlowPaths for diverging areas, 2=single flowpaths (ne
$outpath//qdsr//$grid//.//$code//$year $once_per_interval
$outpath//qisr//$grid//.//$code//$year $once_per_interval
$outpath//qbsr//$grid//.//$code//$year $once_per_interval
$outpath//qgsr//$grid//.//$code//$year $once_per_interval
$outpath//$surfspeed_grid    # grid with actual flow velocity of surface flow in m/s
$Writegrid                   # writegrid for this grid
$outpath//$surfflux_grid     # grid with actual flow amounts of surface flow in m^3/s
$Writegrid              # writegrid for this grid
0.001                   # maximum wake lenght iteration difference (if Delta_A_nl < this value, i
40                      # maximum number of iterations for a_NL
0.0001                  # maximum flow velocity iteration difference (if Delta v is less than thi
40                      # maximum number of iterations for v
```

```
30                      # shortest sub-time step in seconds
3600                    #longest allowed sub time step (even if flow travel times are longer, the
0.02                    # minimum water depth for regarding roughenss of crops in m (shallower sh
2.0                     # ConcentrationFactor takes into account the micro scale concentration of
$readgrids              # readgrid code 0 do not read, 1 = read grids
$outpath//sfstsr//$grid//.//$code//$year $once_per_interval

[lake_model]
1                           # 0=ignore this module, 1 = run the module
2                           # method for recalculating DHM,
# 1 = do not change the DHM, it refects already the ground surface of the lakes,
# 2 = use mean_pond_grid to calculate dhm corrections
# max_pond_grid will be used for mapping the cells pond content to a lake during model runs
0.1 # Albedo_OpenWater (will be used only, when the pond is filled with water
0.4  # z0 for water (usage as above)
$readgrids                  # readgrid code 0 do not read, 1 = read grids -->
# if 0, the initial valte for the POND-grid as Volume of Lakes and Reservoirs is set by V0 from ...
# if readgrids=1, no initialization in done (POND-Grid is read in) but the Vakt-Value is set by...

[unsatzon_model]
1       # 0=ignore this module, 1 = run the module
$time   # duration of a time step in minutes
3       # method, 1=simple method (not supported any more), 2 = FDM-Method 3 = FDM-Met
1       # interaction with surface water: 0 = no , 1 = exfiltration possible 2 = in+exfiltration
1       # surface storage in ponds: 0 = no ponds, 1 = using ponds for surface
0       # artificial drainage: 0 = no artificial drainage 1 = using drainage (d
0       # clay layer: 0 = no clay layer,  1 = assuming a clay layer in
5e-8     # permeability of the clay layer (is used for the clay layer only)
4       # parameter for the initialization of the gw_level (range between 1..levels (standard: 4))
$outpath//qdra//$grid//.//$code//$year $sum_per_24Invs
$outpath//gwst//$grid//.//$code//$year $avrg_per_24Invs
$outpath//gwn_//$grid//.//$code//$year $sum_per_24Invs
$outpath//sb05//$grid//.//$code//$year $avrg_per_24Invs
$outpath//sb1_//$grid//.//$code//$year $avrg_per_24Invs
$outpath//wurz//$grid//.//$code//$year $avrg_per_24Invs
$outpath//infx//$grid//.//$code//$year $sum_per_24Invs
$outpath//pond//$grid//.//$code//$year $avrg_per_24Invs
$outpath//qdir//$grid//.//$code//$year $once_per_interval
$outpath//qifl//$grid//.//$code//$year $once_per_interval
$outpath//qbas//$grid//.//$code//$year $once_per_interval
$outpath//qges//$grid//.//$code//$year $once_per_interval
$outpath//gwin//$grid//.//$code//$year $once_per_interval
$outpath//gwex//$grid//.//$code//$year $once_per_interval
$outpath//macr//$grid//.//$code//$year $sum_per_24Invs
$outpath//qinf//$grid//.//$code//$year $once_per_interval
$outpath//$SB_1_grid                        # grid with actual soil water content for the
$Writegrid                          # Writecode for this grid
$outpath//$SB_2_grid                        # grid with actual soil water content for the entire
unsaturat
$Writegrid                          # Writecode for this grid
$outpath//$ROOTgrid                         # grid with root depth
$Writegrid                                  # Writecode for this grid
$outpath//$Thetastack                       # stack, actual soil water content for all soil levels
$Writegrid                                  # Writecode for this stack
$outpath//$hydraulic_heads_stack    # stack, contaiing hydraulic heads
$Writestack                                 # Writecode for this stack
$outpath//$geodetic_altitude_stack  # stack, containig geodaetic altitudes of the soil levels (low
$Writestack                                 # Writecode for this stack
$outpath//$flowstack                        # stack, containing the outflows from the soil levels
$Writestack                                 # Writecode for this stack
$outpath//$GWdepthgrid                      # grid with groudwaterdepth
$Writegrid                                  # Writecode for this grid
$outpath//$GWthetagrid                      # grid with theta in GWLEVEL
$Writegrid                                  # Writecode for this grid
$outpath//$GWNgrid                          # grid with groundwater recharge
$Writegrid                                  # Writecode for this grid
$outpath//$GWLEVELgrid                      # grid with level index of groundwater surface (Index
der Schi
$Writegrid                                  # Writecode for this grid
$outpath//$QDRAINgrid                       # grid with the drainage flows
$Writegrid                                  # Writecode for this grid
$outpath//$SATTgrid                         # grid with code 1=saturation at interval start, 0 no
sat.
$Writegrid                                  # Writecode for this grid
$outpath//$INFEXgrid                        #  grid  with  infiltration  excess  in  mm  (surface
discharge)
$Writegrid                                  # Writecode for this grid
$outpath//$QDgrid                           # grid with direct discharge
1//$Writegrid                       # Writecode for this grid
$outpath//$QIgrid                           # grid with Interflow
```

```
1//$Writegrid                          # Writecode for this grid
$outpath//$QBgrid                             # grid with baseflow
1//$Writegrid                          # Writecode for this grid
$outpath//$GWINgrid                           # grid with infiltration from rivers into the soil
(groundwate
1//$Writegrid                          # Writecode for this grid
$outpath//$GWEXgrid                           # grid with exfiltration (baseflow) from groundwater
(is only
1//$Writegrid                          # Writecode for this grid
$outpath//$act_pond_grid                      # grid with content of ponding storge
$Writegrid                                    # Writecode for this grid
$outpath//$UPRISEgrid                         # grid with amount of capillary uprise (mm)
1//$Writegrid                          # Writecode for this grid
$outpath//$PERCOLgrid                         # grid with amount of percolation (mm)
1//$Writegrid                          # writegrid for this grid
$outpath//$MACROINFgrid                       # grid with amount of infiltration into macropores (mm)
1//$Writegrid                          # Writecode for this grid
$outpath//$irrig_grid                         # grid with irrigation amount (will be written when
irrigation
$Writegrid                                    # writegrid for this grid (however: will be written
when irrig
70 50                          # coordinates of control plot, all theta and qu-values are written to
$outpath//qbot//$grid//.//$code//$year        # name of a file containing the flows between the laye
$outpath//thet//$grid//.//$code//$year        # name of a file containing the soil moisture as theta
$outpath//hhyd//$grid//.//$code//$year        # name of a file containing the hydraulic head of the
$outpath//otherdata//$grid//.//$code//$year   # name of a file containing some other water balance d
$outpath//etrd//$grid//.//$code//$year        # name of a file containing the withdrawal of soil wat
$outpath//intd//$grid//.//$code//$year        # name of a file containing the interflow for the soil
1       278     # range for subbasin codes
5       5       # kd
10      10      # ki
22      22      # dr
0.4     0.4     # k in qb = Q0 * exp(-k/z) with z = depth to groundwater
0.1     0.1     # Q0 in the above formula
0.15    0.15    # fraction of surface runoff on snow melt
$readgrids                                    # meanings are extended now! read the follwing
$outpath//storage_richards.ftz                # if readgrids = 1, then this file contains th
300                                           # minimum dynamic time step in secounds. the s
$outpath//step//$grid//.//$code//$year $avrg_per_24Invs
$outpath//$SUBSTEPSgrid                                 # grid with number of substeps --> a good idea
$Writegrid                                    # for substeps, the areal distribution is of i

# the following section for heat transfer can be used with WaSiM version 9.0 ff
# it is no
# expected input grids (as standard- or variable grids) are:
# depth of lower boundary condition (if omitted, the lower soil model boundary is taken as default)
# temperatur at lower boundary (if omitted: a mean annual temperature must be defined
# in the following parameter list. The lower boundary condition is then taken from this
# value (taking into account -0.6°K per 100m elevation increase), see parameters below

[heat_transfer]
1                                             # 0 = do not model heat transfer, 1 = heat transfer is
modelled
1  #  0=no processing, 1=yes, only heat diffusion, 2 = yes, heat diffusion and advection
# (by infiltrating water, not yet available))
0       #  vertical heat transfer in snow cover (not yet available)
0       #  2D lateral heat transfer by advection (coupled to water transport) in groundwater
# (not yet available)
#parameters
# the lower boundary condition for temperature may either be defined by a grid with the
# internal name _T_Lower_Boundary_Condition_ or created by using the annual temperature
# and the lapse rate as defined in the next two lines
-10.0  # used when no grid "_T_Lower_Boundary_Condition_" was read in only: mean annual air
# temperature reduced to sea level to be used as lower boundary condition (e.g. 5°C) --> used for
# definition of the lower boundary condition at lower soil boundary, if no grid with lower boundary
# condition was read in
-0.007 # used when no grid "_T_Lower_Boundary_Condition_" was read in only: temperature gradient
# (e.g. -0.007 K/m) for defining the lower boundary condition (used if no grid with lower boundary
# condition was found)
# default soil "constants": can be changed in the soil table (using DryHeatCapacity, DryDensity and
# DryThermalConduct as parameter names)
800     # default heat capacity of dry soil in J/(Kg*K), default 800 --> value may be given in
# detail for each soil type in the soil table
1500    # default density of dry soil in Kg/m^3 , default 1500 --> value may be given in detail for
# each soil type in the soil table
0.58    # default thermal conductivity for dry soil in J/(m*s*K) or W/(m*K): default: 0.58 --> value
# may be given in detail for each soil type in the soil table
1e-12    # reduced k_sat (minimum hydraulic conductivity for fully frozrn soils)
# thermodynamic constants of water and ice (not for calibration! these are constants giving only
# marginal room for variations)
```

```
0.5562 # thermal conductivity of liquid water
2.33   # thermal conductivity of ice (0°C...-20°C)
4187   # heat capacity of water in J/(Kg*K)
1940   # heat capacity of ice at -20°C in J/(Kg*K)
2090   # heat capacity of ice at 0°C in J/(Kg*K)
334000 # latent heat of freezing in J/Kg
1000   # density of water in Kg/m^3
# other parameters (not for calibrating, but there is no clear lioterature value)
1.22   # scaling factor (solution of the clapeyron equation, literature gives values of 1.8 up to
# 123, but this may be measure dependent. Theoretical value is dH/T_m = 1.22 J/(Kg*K))
0.9    # SE value which must be underrun to evaluate the soil layer as frozen for the
# Thawdepth-output grid and statistics
1800   # minimum sub time step allowed for heat transfer model (if the required time step would be
# shorter, numeric errors like extrem temperature fluctuations are possible)
1800   # maximum sub time step allowed for heat transfer model (to avoid instabilites induced by
# the nonlinearity of the processes)
1.0    # n-factor for freezing (factor applied to the air temperature to get the temperature at the
# soil surface as upper boundary condition when temperatures are negative
1.0    # n-factor for thawing (factor applied to the air temperature to get the temperature at the
# soil surface as upper boundary condition when temperatures are positive
# this value ranges from 0.01 to 0.99 with 0.01 defining beginning freezing (1% ice) and 0.99
# defining complete freezing (99% ice, only smallest pores may contain water
# output grids and statistics
$outpath//ts_loc//$grid//.//$code//$year # results soil temperature for control point
$outpath//ts_avg//$grid//.//$code//$year $once_per_interval # soil temperature thaw depth
$outpath//$Temperaturestack                 # stack, actual soil water content for all soil levels
$Writegrid                                  # Writecode for this stack
$outpath//$ThawDepthGridTMod                #  active layer thickness relative to the soil
surface
# (deepest thawing front in the soil profile) → result grid
$Writegrid                                  # Writecode for this stack
$readgrids

[ExternalCoupling]
0                                           # 0 = no coupling, 1=coupling

[irrigation]
0                                           # 0=ignore this module, 1 = run the module
$time                                       # duration of a time step in minutes
$outpath//irgw//$grid//.//$code//$year $once_per_interval   # statistic of the irrigation water fr
$outpath//irsw//$grid//.//$code//$year $once_per_interval   # statistic of the irrigation water fr

[groundwater_flow]
1                                           # 0=ignore the module, 1 = run the module
$time                                       # duration of a time step in minutes; doen't c
1                                           # solving method: 1=Gauss-Seidel-iteration (us
1000                                        # if iterative solving method (1): max.numbero
0.0001                                      # if iterative solving method (1): max. change
0.0                                         # Alpha for estimation of central differences
-1.20                                       # factor for relaxing the iteration if using i
$readgrids                                  # 1=read grids for heads from disk, 0=do not r
1                                           # number of layers
70 50                                       # coordinates of a control point for all fluxe
$outpath//glog//$grid//.//$code//$year      # name of a file containing the flows between
1                                           # use Pond Grid -> this enables the model to u
$outpath//$head1grid                        # (new) grid for hydraulic heads for layer 1
$Writegrid                                  # writecode for hydraulic heads for layer 1
$outpath//$flowx1grid                       # (new) grid for fluxes in x direction for lay
$Writegrid                                  # writecode for flux-x-grid in layer 1
$outpath//$flowy1grid                       # (new) grid for fluxes in y direction for lay
$Writegrid                                  # writecode for flux-y-grid in layer 1
$outpath//$GWbalance1grid                   # (new) grid for balance (difference of storag
$Writegrid                                  # writecode for balance control grid in layer


# this paragraph is not needed for WaSiM-uzr but for the WaSiM-version with the variable saturated
area approach (after Topmodel)
[soil_model]
0                                           # 0=ignore this module, 1 = run the module
$time

[routing_model]
1                                                   #  0=ignore  this  module,  1  =  run  the
module, 2=run the module with observed inflows int
$time
1 2400 90 24                                        #  minimum/maximum  specific  discharge
(l/s/km^2), number of log. fractions of the range
$outpath//qgko//$grid//.//$code//$year $routing_code
$inpath_hydro//spen9610.dat
13                                          # number of following column descripotr (which
```

```
column in the spec. disch. file corresp
1      1                                              # first number: subbasin, second: column index
2      2
3      3
4      4
5      5
6      6
7      7
8      8
9      9
11     11
12     12
13     13
15     10
48    #  timeoffset
TG 102 (AE=15.350, AErel=1.0)
 from AL 1 (modus = intern 0.0 1.0 10 m^3/s )
TG 103 (AE=16.330, AErel=1.0)
 from AL 2 (modus = intern 0.0 1.0 10 m^3/s )
TG 104 (AE=17.090, AErel=1.0)
 from AL 3 (modus = intern 0.0 1.0 10 m^3/s )
TG 106 (AE=11.01, AErel=1.0)
 from AL 4 (modus = intern 0.0 1.0 10 m^3/s )
TG 107 (AE=4.06, AErel=1.0)
 from AL 5 (modus = intern 0.0 1.0 10 m^3/s )
TG 112 (AE=5.750, AErel=1.0)
 from AL 6 (modus = intern 0.0 1.0 10 m^3/s )
TG 113 (AE=2.420, AErel=1.0)
 from AL 7 (modus = intern 0.0 1.0 10 m^3/s )
TG 114 (AE=4.090, AErel=1.0)
 from AL 8 (modus = intern 0.0 1.0 10 m^3/s )
TG 115 (AE=6.930, AErel=1.0)
 from AL 9 (modus = intern 0.0 1.0 10 m^3/s )
TG 116 (AE=2.620, AErel=1.0)
etc. defining all routing descriptions for head water basins witch abstraction are defined for
# now the regular routing descriptions follow
TG 277 (AE=6.25, AErel=1.0)
 from AL 88 (modus = extern ableitungSB277.dat 0.0 1 4.0 m^3/s)
TG 278 (AE=8.25, AErel=1.0)
 from AL 87 (modus = extern ableitungSB278.dat 0.0 1 4.0 m^3/s)
TG 10 (AE=135.250, AErel=1.0)
 from OL 233 (kh=0.1, kv=0.4, Bh= 2.0, Bv=  8.0, Th= 0.42, Mh=25.0, Mv=10.0, I=0.0449, L=20071.0,
AE=10.000)
  and OL 232 (kh=0.1, kv=0.4, Bh= 2.0, Bv=  7.9, Th= 0.42, Mh=25.0, Mv=10.0, I=0.0446, L=20278.1,
AE=9.500)
  and OL 278 (kh=0.1, kv=0.4, Bh= 1.5, Bv=  5.9, Th= 0.40, Mh=25.0, Mv=10.0, I=0.0734, L=12071.0,
AE=8.250) # is an abstraction to the outside
  and OL 277 (kh=0.1, kv=0.4, Bh= 1.4, Bv=  5.6, Th= 0.36, Mh=25.0, Mv=10.0, I=0.0675, L=10449.7,
AE=6.250) # is an abstraction to the outside
  and SP 15  ( file = $outpath//spv___15_010.//$code//$year , V0 = 2.0E5, C0 = 0.0  0.0  0.0  0.0
0.0  0.0  0.0  0.0  0.0, dTmin = 60, UseRunningAvg = 0)
  and AL 78  (modus = intern_with_rule_from_reservoir )
TG 15 (AE=135.50, AErel=1.0)
 from OL 10  (kh=0.1, kv=0.4, Bh= 2.0, Bv=  8.0, Th= 0.42, Mh=25.0, Mv=10.0, I=0.0449, L=100.0,
AE=135.25)
   and ZL 78   (modus = intern , kh=0.4, kv=0.4, Bh=10.0, Bv=60.0, Th=1.4, Mh=27.0, Mv=15.0,
I=0.0084, L=10000, AE=135.5) #from virtual reservoi
TG 13 (AE=38.000, AErel=1.0)
 from OL 272 (kh=0.1, kv=0.4, Bh= 1.3, Bv=  5.3, Th= 0.36, Mh=25.0, Mv=10.0, I=0.0759, L=3621.3,
AE=6.250)
  and AL 85  (modus = extern ableitungSB13.dat 0.1 1 0.08 m^3/s)   # Ableitung laut hydrologischem
Atlas von 0.08 m3/s
TG 270 (AE=18.75, AErel=1.0)
 from SP 13  ( file = $outpath//spv___13_270.//$code//$year , V0 = 18.0E6, C0 = 0.0  0.0  0.0  0.0
0.0  0.0  0.0  0.0  0.0, dTmin = 60, UseRunningAvg = -1)
   and ZL 73   (modus = intern , kh=0.4, kv=0.4, Bh=10.0, Bv=60.0, Th=1.4, Mh=27.0, Mv=15.0,
I=0.0084, L=1000, AE=38.50 )
  and AL 86  (modus = extern ableitungSB08.dat 0.1 1.0 1.2 m^3/s)
             # Ableitung laut hydro
  and AL 74  (modus = intern_with_rule_from_reservoir ) # for power generation
TG 8 (AE=203.000, AErel=1.0)
 from OL 13  (kh=0.1, kv=0.4, Bh= 3.9, Bv= 15.7, Th= 0.69, Mh=25.0, Mv=10.0, I=0.0311, L=14313.6,
AE=38.000)
  and OL 271 (kh=0.1, kv=0.4, Bh= 6.3, Bv= 25.1, Th= 0.68, Mh=25.0, Mv=10.0, I=0.0115, L=11692.3,
AE=38.500)
  and OL 269 (kh=0.1, kv=0.4, Bh= 1.6, Bv=  6.4, Th= 0.48, Mh=25.0, Mv=10.0, I=0.0889, L=3621.3,
AE=12.750)
  and OL 270 (kh=0.1, kv=0.4, Bh= 2.1, Bv=  8.6, Th= 0.46, Mh=25.0, Mv=10.0, I=0.0472, L=11985.2,
AE=12.750)
   and ZL 74   (modus = intern , kh=0.4, kv=0.4, Bh=10.0, Bv=60.0, Th=1.4, Mh=27.0, Mv=15.0,
```

```
I=0.0084, L=5000, AE=42.00 )

… (some routing descriptions are cut here)

TG 1 (AE=5312.500, AErel=1.0)
 from OL 15   (kh=0.2, kv=0.4, Bh=13.8, Bv= 55.3, Th= 1.38, Mh=25.0, Mv=10.0, I=0.0027, L=8242.6,
AE=135.50)
   and OL  2   (kh=0.2, kv=0.4, Bh=49.2, Bv=196.7, Th= 4.92, Mh=25.0, Mv=10.0, I=0.0024, L=37556.2,
AE=3770.000)
  and OL 229 (kh=0.2, kv=0.4, Bh= 2.7, Bv= 10.6, Th= 0.48, Mh=25.0, Mv=10.0, I=0.0325, L=28642.0,
AE=14.250)
  and OL 228 (kh=0.2, kv=0.4, Bh= 2.5, Bv= 10.1, Th= 0.47, Mh=25.0, Mv=10.0, I=0.0339, L=30056.2,
AE=13.250)
  and OL 231 (kh=0.2, kv=0.4, Bh= 6.6, Bv= 26.3, Th= 0.94, Mh=25.0, Mv=10.0, I=0.0207, L=20227.8,
AE=90.750)
  and OL 227 (kh=0.2, kv=0.4, Bh= 4.1, Bv= 16.2, Th= 0.41, Mh=25.0, Mv=10.0, I=0.0095, L=22434.9,
AE=9.750)
  and OL 226 (kh=0.2, kv=0.4, Bh= 4.0, Bv= 16.0, Th= 0.43, Mh=25.0, Mv=10.0, I=0.0113, L=27349.1,
AE=11.250)
  and OL 225 (kh=0.2, kv=0.4, Bh= 2.8, Bv= 11.3, Th= 0.53, Mh=25.0, Mv=10.0, I=0.0350, L=36263.3,
AE=18.500)
  and OL 224 (kh=0.2, kv=0.4, Bh= 2.1, Bv=  8.4, Th= 0.38, Mh=25.0, Mv=10.0, I=0.0331, L=41505.9,
AE=7.750)
   and OL  6   (kh=0.2, kv=0.4, Bh=25.7, Bv=102.7, Th= 2.57, Mh=25.0, Mv=10.0, I=0.01, L=40763.3,
AE=676.000)
  and OL 222 (kh=0.2, kv=0.4, Bh= 8.5, Bv= 34.1, Th= 1.06, Mh=25.0, Mv=10.0, I=0.0154, L=41920.1,
AE=125.000)
   and ZL 66  (modus = intern , kh=0.2, kv=0.4, Bh=20.0, Bv=60.0, Th=2.4, Mh=25.0, Mv=15.0, I=0.01,
L=20000, AE=18.50 )
  and ZL 67  (modus = intern , kh=0.2, kv=0.4, Bh=20.0, Bv=60.0, Th=2.4, Mh=25.0, Mv=15.0, I=0.01,
L=25000, AE=28.00 )
  and ZL 80  (modus = intern , kh=0.2, kv=0.4, Bh=20.0, Bv=60.0, Th=2.4, Mh=25.0, Mv=15.0, I=0.01,
L=25000, AE=28.00 )
TG 99 (AE=5355.500, AErel=1.0)
 from OL  1   (kh=0.2, kv=0.4, Bh=64.1, Bv=256.6, Th= 6.41, Mh=25.0, Mv=10.0, I=0.0011, L=5707.1,
AE=5306.500)

# abstration rules are defined this way:
# first row: number of following columns, followed by the julian days for which rules will be
# established
# the Julian day describes 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
# other rows: discharge (m^3/s), followed by the abstraction valid for this discharge (m^3/s)
# or reservoir volume in m^3, followed by the abstraction in m^3/s --> to differentiate between
# discharge in m^3/s and reservoir content in m^3, the keyword "modus = intern_with_rule" must
# be extended by the keyword "_from_reservoir", i.e. intern_with_rule_from_reservoir

[abstraction_rule_abstraction_66]
4       10      70      91      121     152     182     213     244     274     320     356     366
0       0       0       0       0       0       0       0       0       0       0       0
5.2e6   3.0     5.0     3.0     3.0     3.0     3.0     3.0     3.0     3.0     3.0     4.0     3.0
26e6    3.0     5.0     3.0     3.0     3.0     3.0     3.0     3.0     3.0     3.0     4.0     3.0
28e6    4.8     4.8     4.8     4.8     4.8     4.8     4.8     4.8     4.8     4.8     4.8     4.8
TargetCap =     360     360     360     360     360     360     360     360     360     360     360
        360
WeekDays  =     12345   12345   12345   12345   12345   12345   12345   12345   12345   12345   12345
        12345
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_67]
4       10      70      91      121     152     182     213     244     274     320     356     366
0       0       0       0       0       0       0       0       0       0       0       0
45e6    18      29      18      18      16      10      10      10      10      18      29      18
215e6   18      29      18      18      16      10      10      10      10      18      29      18
227e6   29      29      29      29      29      29      29      29      29      29      29      29
TargetCap =     360     360     360     360     360     360     360     360     360     360     360
        360
WeekDays  =     12345   12345   12345   12345   12345   12345   12345   12345   12345   12345   12345
        12345
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

etc. for every abstraction defined in the routing descriptions above

# the reservoirs must have an abstraction rule as well, even if they have no abstraction at all or
a constant abtraction (Restwassermenge) onl
```

```
#if a capacity for the flood

[abstraction_rule_reservoir_1] # AE ca 18.5 km^2, flood-spill: 15 m^3/s,
3       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
25.99e06 0      0       0       0       0       0       0       0       0       0       0       0
40.00e06 15     15      15      15      15      15      15      15      15      15      15      15

[abstraction_rule_reservoir_2] #  AE ca 28 km^2 incl. inflows: 146.5 km^2, flood spill: 60 m^3/s,
3       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
224.99e6 0      0       0       0       0       0       0       0       0       0       0       0
230e6   60      60      60      60      60      60      60      60      60      60      60      60

etc. for every reservoir one rule


# declaring some common variables for vegetation period dependent grid-writing
# default (if not used in land use table at all) is JDVegReset = 1 and JDVegWrite = 365
$set $JDVegReset = 1
$set $JDVegWrite = 365

[multilayer_landuse]
16 # count of multilayer landuses
1  Wasserflächen      { Landuse_Layers = 16, -9999; k_extinct = 0.3; LAI_scale = 20;}
2  Bebauung           { Landuse_Layers = 2,  -9999; k_extinct = 0.3; LAI_scale = 20;}
3  Teilbebauung               { Landuse_Layers = 1,  -9999; k_extinct = 0.3; LAI_scale = 20;}
4  Wald               { Landuse_Layers = 13,  4;    k_extinct = 0.3; LAI_scale = 20;}
5  offener_Wald               { Landuse_Layers = 14,  6;    k_extinct = 0.3; LAI_scale = 20;}
6  Gebueschwald               { Landuse_Layers = 10,  4;    k_extinct = 0.3; LAI_scale = 20;}
7  Ackerland          { Landuse_Layers = 7,  -9999; k_extinct = 0.3; LAI_scale = 20;}
8  Gruenland          { Landuse_Layers = 5,  -9999; k_extinct = 0.3; LAI_scale = 20;}
9  Obstbauflaechen    { Landuse_Layers = 17,  4;    k_extinct = 0.3; LAI_scale = 20;}
10 Gebuesch           { Landuse_Layers = 10,  -9999;       k_extinct = 0.3; LAI_scale = 20;}
11 verbuschtes_Weidel { Landuse_Layers = 9,  -9999; k_extinct = 0.3; LAI_scale = 20;}
12 uebrige_Gruenfl    { Landuse_Layers = 6,  -9999; k_extinct = 0.3; LAI_scale = 20;}
13 Moore_Suempfe      { Landuse_Layers = 15,  -9999;       k_extinct = 0.3; LAI_scale = 20;}
14 versteinte_wiesen  { Landuse_Layers = 4,  -9999; k_extinct = 0.3; LAI_scale = 20;}
15 Felsflaechen_Oedland       { Landuse_Layers = 3,  -9999; k_extinct = 0.3; LAI_scale = 20;}
16 Gletscher          { Landuse_Layers = 22,  -9999;       k_extinct = 0.3; LAI_scale = 20;}


[landuse_table]
19
1  teilversiegelte_Flaechen {method = VariableDayCount;
               RootDistr        = 1.0;
               TReduWet         = 0.95;
               LimitReduWet     = 0.5;
               HReduDry         = 3.5;
               IntercepCap      = 0.3;
JulDays          = 15    46    74    105   135   166    196    227    258    288    319    349  ;
Albedo           = 0.2   0.2   0.2   0.2   0.2   0.2    0.2    0.2    0.2    0.2    0.2    0.2  ;
rsc              = 100   100   100   80    70    70     70     70     70     70     100    100  ;
rs_interception  = 0.5   0.5   0.5   0.5   0.5   0.5    0.5    0.5    0.5    0.5    0.5    0.5  ;
rs_evaporation   = 150   150   150   150   150   150    150    150    150    150    150    150  ;
LAI              = 1     1     1     1     1     1      1      1      1      1      1      1    ;
Z0               = 1     1     1     1     1     1      1      1      1      1      1      1    ;
VCF              = 0.5   0.5   0.5   0.5   0.5   0.5    0.5    0.5    0.5    0.5    0.5    0.5  ;
RootDepth        = 0.4   0.4   0.4   0.4   0.4   0.4    0.4    0.4    0.4    0.4    0.4    0.4  ;
AltDep           = 0.025 0.025 0.025 0.025 0.025 0.025 -0.025 -0.025 -0.025 -0.025 -0.025 -0.025;
}
2  versiegelte_Flaechen {method = VariableDayCount;
               RootDistr        = 1.0;
               TReduWet         = 0.95;
               LimitReduWet     = 0.5;
               HReduDry         = 3.5;
               IntercepCap      = 0.2;
JulDays          = 15    46    74    105   135   166    196    227    258    288    319    349  ;
Albedo           = 0.2   0.2   0.2   0.2   0.2   0.2    0.2    0.2    0.2    0.2    0.2    0.2  ;
rsc              = 100   100   100   100   100   100    100    100    100    100    100    100  ;
rs_interception  = 0.5   0.5   0.5   0.5   0.5   0.5    0.5    0.5    0.5    0.5    0.5    0.5  ;
rs_evaporation   = 200   200   200   200   200   200    200    200    200    200    200    200  ;
LAI              = 1     1     1     1     1     1      1      1      1      1      1      1    ;
Z0               = 1     1     1     1     1     1      1      1      1      1      1      1    ;
VCF              = 0.2   0.2   0.2   0.2   0.2   0.2    0.2    0.2    0.2    0.2    0.2    0.2  ;
RootDepth        = 0.4   0.4   0.4   0.4   0.4   0.4    0.4    0.4    0.4    0.4    0.4    0.4  ;
AltDep           = 0.025 0.025 0.025 0.025 0.025 0.025 -0.025 -0.025 -0.025 -0.025 -0.025 -0.025;
}
3  vegetationslose_Flaechen {method  = VariableDayCount;
               RootDistr        = 1.0;
```

```
               TReduWet         = 0.95;
               LimitReduWet     = 0.5;
               HReduDry         = 3.5;
               IntercepCap      = 0.2;
JulDays          = 15    46    74    105   135   166    196    227    258    288    319    349  ;
Albedo           = 0.2   0.2   0.2   0.2   0.2   0.2    0.2    0.2    0.2    0.2    0.2    0.2  ;
rsc              = 100   100   100   100   100   100    100    100    100    100    100    100  ;
rs_interception  = 0.5   0.5   0.5   0.5   0.5   0.5    0.5    0.5    0.5    0.5    0.5    0.5  ;
rs_evaporation   = 100   100   100   100   100   100    100    100    100    100    100    100  ;
LAI              = 0.5   0.5   0.5   0.5   0.5   0.5    0.5    0.5    0.5    0.5    0.5    0.5  ;
Z0               = 0.1   0.1   0.1   0.1   0.1   0.1    0.1    0.1    0.1    0.1    0.1    0.1 ;
VCF              = 0.2   0.2   0.2   0.2   0.2   0.2    0.2    0.2    0.2    0.2    0.2    0.2  ;
RootDepth        = 0.1   0.1   0.1   0.1   0.1   0.1    0.1    0.1    0.1    0.1    0.1    0.1  ;
AltDep           = 0.025 0.025 0.025 0.025 0.025 0.025 -0.025 -0.025 -0.025 -0.025 -0.025 -0.025;
}
etc. for every land use code one section with specific parameters
22  Eisflaechen  {method  = VariableDayCount;
              RootDistr      = 1;
              TReduWet       = 1;
              LimitReduWet   = 1;
              HReduDry       = 150;
              IntercepCap    = 0;
              JulDays        = 365;
              Albedo         = 0.38;
              rsc            = 200;
              rs_interception = 0;
              rs_evaporation        = 400;
              LAI            = 0.1;
              Z0             = 0.01;
              VCF            = 0.1;
              RootDepth      = 0.1;
              AltDep         = 0;
               }
23  Firnflaechen  {method  = VariableDayCount;
              RootDistr      = 1;
              TReduWet       = 1;
              LimitReduWet   = 1;
              HReduDry       = 150;
              IntercepCap    = 0;
              JulDays        = 365;
              Albedo         = 0.60;
              rsc            = 200;
              rs_interception = 0;
              rs_evaporation        = 400;
              LAI            = 0.1;
              Z0             = 0.01;
              VCF            = 0.1;
              RootDepth      = 0.1;
              AltDep         = 0;
               }



$set $e3 = e-2
$set $e4 = e-3
$set $e5 = e-4
$set $e6 = e-5
$set $e7 = e-6
$set $e8 = e-7
$set $e9 = e-8
$set $e10 = e-9


[soil_table]
51
#co- name of the      # van Genuchten Parameter nach Carsel & Parrish (1988)
#de  soil profile
#-- ---------------
23 1_Seen_subhydrische_Boeden {method = MultipleHorizons;   # AE = 0.28 %
                      PMacroThresh   = 0 ;
                      MacroCapacity  = 0 ;
                      CapacityRedu   = 0 ;
                      MacroDepth     = 0 ;
                       horizon       = 1       2     3     ;
                      Name           = SIL     SIL    CL     ;
                      ksat           = 3//$e7          1//$e7   1//$e7   ; # SIL: 1//$e6
                             k_recession    = 0.9     0.9    0.9    ;
                       theta_sat     = 0.45    0.45   0.40   ;
                       theta_res     = 0.067   0.067  0.095  ;
                       alpha         = 2.00    2.0    1.90   ;
```

```
                        Par_n         = 1.41    1.41   1.31   ;
                        Par_tau       = 0.5     0.5    0.5    ;
                        thickness     = 0.10    0.30   2.00   ;     # 20.0 m
                        layers        = 4       2      9      ;
                       }
26 6_Moorboeden {method = MultipleHorizons;  #vgl. F1       # AE = 0.01 %
                       PMacroThresh   = 1.0 ;
                       MacroCapacity  = 20.0 ;
                       CapacityRedu   = 1.0 ;
                       MacroDepth     = 2.0 ;
                        horizon       = 1       2      3       ;
                       Name           = HMT     NMT    hHr_SL3 ;
                       ksat           = 1//$e5          3//$e6    1//$e6    ;
                              k_recession    = 0.9     0.9     0.9   ;
                        theta_sat     = 0.65    0.55    0.40    ;
                        theta_res     = 0.30    0.30    0.065   ;
                        alpha         = 4.0     4.0     7.50    ;
                        Par_n         = 1.2     1.2     1.89    ;
                        Par_tau       = 0.5     0.5     0.5     ;
                        thickness     = 0.10    0.30    2.00    ;
                        layers        = 4       2       9       ;
                       }
13 7_Siedlungsflaechen {method = MultipleHorizons; #vgl. 8  # AE = 0.42 %
                       PMacroThresh   = 4.0 ;
                       MacroCapacity  = 4.0 ;
                       CapacityRedu   = 1.0 ;
                       MacroDepth     = 2.0 ;
                        horizon       = 1       2     3      ;
                       Name           = SIL     SIL   R      ;
                       ksat           = 1//$e6          3//$e7   5//$e8   ; # SIL: 1//$e6
                              k_recession    = 0.9     0.9    0.9    ;
                        theta_sat     = 0.45    0.45   0.20   ;
                        theta_res     = 0.067   0.067  0.04   ;
                        alpha         = 2.0     2.0    8.0    ;
                        Par_n         = 1.41    1.41   1.80   ;
                        Par_tau       = 0.5     0.5    0.5    ;
                        thickness     = 0.10    0.30   2.00   ;
                        layers        = 4       2      9      ;
                       }
91 8_Felsflaechen {method = MultipleHorizons;                       # AE = 37 %
                       PMacroThresh   = 4.0 ;
                       MacroCapacity  = 4.0 ;
                       CapacityRedu   = 1.0 ;
                       MacroDepth     = 2.0 ;
                        horizon       = 1       2     3      4       ;
                       Name           = SIL     CL    R      R       ;
                       ksat           = 1//$e6          3//$e7   1//$e8   1//$e8     ; # SIL: 1//$e6

                              k_recession    = 0.9     0.9    0.9    0.9     ;
                        theta_sat     = 0.45    0.41   0.20   0.20    ;
                        theta_res     = 0.067   0.095  0.04   0.04    ;
                        alpha         = 2.00    1.90   8.00   8.00    ;
                        Par_n         = 1.41    1.31   1.80   1.80    ;
                        Par_tau       = 0.5     0.5    0.5    0.5     ;
                        thickness     = 0.20    0.20   0.30   0.40    ;
                        layers        = 4       4      4      3      ;

                       }
etc. until all 51 soils are defined

[substance_transport]
0

[irrigation_table]
2                  # number of following irrigation codes, per row one use
#
#Code name         method     from        control by
#                  (0=no irr, (1=GW       demand:           table:
#                   1=table1,    2=river)    psi[m]                          [mm]              [mm]
#                   2=table2)                start  stop   count  MM1 DD1 amount1 MM2 DD2 amount2 MM3
#                  3=demand                 or
#                  4=ETR/ETP<e_rp           e_rp (method 4 only)
#---- ------------ ---------- --------  --------------  ------ --------------- ---------------  2
 1   LW-Flaeche    4          2           0.8
 1   Restflaeche   0          2           0.8

[special_output]
1 # 0=do not run this module, 1=run this module
$time # duration of a time step in minutes --> only for compatibility here.
numfiles = 2;
```

```
outputfile { header = glacierdata;
        filename = $outpath//special_output_glaciers.//$code//$year;
        entity {
                ID      = GlacierMassBalance;
                Symbol  = GMB;
                Xcoords = 771371, 801115, 771211;
                Ycoords = 214666, 194848, 164323;
        }
        entity {
                ID      = melt_from_firn;
                Symbol  = Mfirn;
                Xcoords = 771371, 801115, 771211;
                Ycoords = 214666, 194848, 164323;
        }
        entity {
                ID      = melt_from_ice;
                Symbol  = Mice;
                Xcoords = 771371, 801115, 771211;
                Ycoords = 214666, 194848, 164323;
        }
}
outputfile { header = soildata;
        filename = $outpath//special_output_soilmodel.//$code//$year;
        entity {
                ID      = theta;
                Symbol  = th;
                Xcoords = 748503, 748503, 748503, 748503, 770572, 770572, 770572, 770572, 770572;
                Ycoords = 196127, 196127, 196127, 196127, 256698, 256698, 256698, 256698, 256698;
                Level   = 1,       2,      3,       4,      1,       2,       3,     4,        5;
        }
        entity {
                ID      = hydraulic_heads;
                Symbol  = hh;
                Xcoords = 748503, 748503, 748503, 748503, 770572, 770572, 770572, 770572, 770572;
                Ycoords = 196127, 196127, 196127, 196127, 256698, 256698, 256698, 256698, 256698;
                Level   = 1,       2,      3,       4,      1,       2,       3,     4,        5;
        }
}
```

7.2 Example of a control file for Tanalys

Tanalys uses a similarly structured control file like WaSiM. All information required for the topographic analysis are contained in this control file. There are sections for each task, enclosed in “[brackets]”. Comments are allowed everywhere if the first character of a row is a “#”. Other comments are allowed behind the parameter entries, but they should also start with a hash-symbol “#”. The command “$set $identifier = $<identifier_value>” may be used to define at maximum 200 symbols.

The digital elevation model as the original data source is read in an own section. All other sections require the specifications of the input-grid names. Each section can be disabled by the “ignorance” parameter. If a task is disabled, but it’s results will be needed by another task, the required grids are first looked for at the hard disk. If they are not found, TANALYS stops its execution (if the sections are not disabled, then TANALYS will look for the grids internally first, even if they exist on the hard disk already).

If a new basin should be analyzed by TANALYS it is usually sufficient to change the symbols $path, $grid and $threshold, sometimes also the Manning-roughness parameter $Manning and the specific discharge for average floods $specific_discharge should be changed (section [Fliesszeiten]). Then, all sections should be enabled by setting their switch-on/off parameters to 1. If TANALYS is started with such a control file, a complete topographical analysis should be performed generating a series of output grids which may be viewed using ShowGrids.

In the section for delineating subbasins ([Einzugsgebiete]), any grids containing pour points may be specified. By default, this are the mouths of rivers of different orders, e.g. rivers of order 2, 3, 4 and 5. However, if the user would like TANALYS to delineate subbasins for given pour points, e.g. for gauge-coordinates, such grids may also be specified in this section. If all the other grids exist, it is also possible to enable only the section [Einzugsgebiete]. Then, the flow directions and flow times

are read from the disk. The pour point grid may also be edited using ShowGrids in order to set gauges to well defined grid cells. pour points must be unique and in the range of 1…32767.

TANALYS may require a lot of computer memory, depending on the size of the grid. Systems like UNIX or LINUX as well as Windows with 64 bit architecture have virtually no memory restrictions, so there should be no practical limit for applying TANALYS to digital elevation models of any size (however, the hardware may still limit the grid size).

```
# control file for topographic analysis
# lines starting with a hash symbol (#) are comments

$set $path       = d:\Data\Input\grids\topo500\ # main path
$set $Grid       = r500  # common grid name
$set $threshold  = 4     # threshold for flow accumulation top create a river
$set $Manning    = 25    # roughness coefficient for flow profile generation
$set $SpecDisc   = 200   # typical specific flood runoff (morphologic effective)

$set $dhm               = //$Grid//.dhm #= dem
$set $rivernetwork            = //$Grid//.lin
$set $Flowdirection           = //$Grid//.fld
$set $Flowaccumulation  = //$Grid//.acc
$set $riverlinks        = //$Grid//.num
$set $StrahlerOrder           = //$Grid//.ord


[Hoehenmodell]          # DEM
1                       # switch 0=do not read in, 1= read in
$Path//$DEM             # path to the DEM file

[Aspect]                # exposition
1                       # switch 0=do not calculate, 1=calculate this section
$Path//$Grid//.exp      # result grid Aspect

[Gefaelle]              # slope
1                       # switch 0=do not calculate, 1=calculate this section
$Path//$Grid//.slp      # result grid slope

[Horizont]              #sky view factor
0                       # switch 0=do not calculate, 1=calculate this section
$Path//$Grid//.hor      # result grid sky view factor
12                           # number of sections to split the circle for
                        # getting the integral sky view factor
                        # (e.g. 12 = every 36° a sample will be taken

#new section for new snow model extensions of WaSiM 9.06.02

[WindCorrectionFactor]  # used for snow correction in snow model of WaSiM
1                 # 0=do not process this section, 1=process this section
$Path//$Grid//.svfdir   # result grid: directed sky view factor for a limited
# sector of a circle (directed svf). Attention: the real wind redistribution
# factor is a grid with the same name but suffix _2 for the extension.
10                # number of sectors to subdivide the limited sector for
# integrating the sky view factor
180               # start azimuth (0=North, 90=East, 180=South etc.)
270               # end azimuth for the directed svf
2.0               # maximum deposition index: 2 means that for an evenly
# distributed terrain the deposition index will vary between 0 and 2 with an
# average of 1, so no effective increase of snow fall will be present. Values
# larger than 2 result in an effective snow fall increase, value less than 2
# (but greater than 0) in a decrease
2000              # elevation offset: the larger this number, the more
# additional snow will even the low elevation cells get since they are handled
```

```
# as if they were higher (by this offset)
0.3               # Minimum remaining snow (shifts the correction factor
# globally)

[SnowRedistribution] # snow redistribution input grids
1
$Path//$Grid//.sd1 # result grid: weighed fraction for flow direction 1 (North)
$Path//$Grid//.sd2 # result grid: weighed fraction for flow direction 2 (West)
$Path//$Grid//.sd3 # result grid: weighed fraction for flow direction 3 (East)
$Path//$Grid//.sd4 # result grid: weighed fraction for flow direction 4 (South)
$Path//$Grid//.eoc # result grid: elevation order (colums)
$Path//$Grid//.eor # result grid: elevation order (rows)
# note: oec and eor grid will be calculated in WaSiM in any case. So the grids
# calculated here are for your reference only

[Profilcurvature]
1                                  # 0 = ignore section, 1 = section is active
$path//$Grid//.prc                 # result grid Profilcurvature

[Plancurvature]
1                                  # 0 = ignore section, 1 = section is active
$path//$Grid//.plc                 # result grid Plancurvature

[Curvature]
1                                  # 0 = ignore section, 1 = section is active
$path//$Grid//.cur                 # result grid Curvature

[Flowdirection]
1                                  # 0 = ignore section, 1 = section is active
$path//$Flowdirection          # result grid flow directions
$path//$Grid//.sin             # result grid sinks
$path//$Grid//.dhk             # result grid DEM with filled up sinks

[Flowaccumulation]
1                                  # 0 = ignore section, 1 = section is active
$path//$Flowaccumulation       # result grid flow accumulation
$path//$Flowdirection          # flow directions (input grid, as created above)
$threshold                     # threshold for creating a river

[Flussnetz] # river links (without coding - only a single value)
1                                  # 0 = ignore section, 1 = section is active
$path//$rivernetwork           # result grid river links (unspecific codes)
$path//$Flowaccumulation       # input grid flow accum. (from [Flowaccumulation])
$threshold                     # threshold for creating a river

[Flussordnungen] # Strahler orders
1                                  # 0 = ignore section, 1 = section is active
$path//$StrahlerOrder          # result grid with Strahler order for rivers
$path//$riverlinks                 # result grid river links, codes by 1..x
$path//$rivernetwork               # input grid riuver network (from
[Flussnetz])
$path//$Flowdirection          # input grid flow directions (from above)

[Fliesskorrektur] #correction of flow directions
1                                  # 0 = ignore section, 1 = section is active
$path//$Grid//.ork             # result grid corrected Strahler orders
$path//$Grid//.nuk             # result grid corrected river codes
$path//$Grid//.flk             # result grid corrected flow directions
$path//$Grid//.ack             # result grid corrected flow accumulation
$threshold
$path//$dhm                        # input grid DEM (from above)
$path//$StrahlerOrder          # input grid river network (Strahler orders)
$path//$Flowdirection          # input grid flow directions (from above)
$path//$Flowaccumulation       # input grid flow accumulation (from above)
```

```
$path//$rivernetwork                # input grid river network (unspecific)
5 # number of correction runs (iterations on different orders)
1     # Strahler order for first correction (is ignored if order does not exist)
2     # Strahler order for 2nd correction
1     # Strahler order for 3rd correction
3     # Strahler order for 4th correction
1     # Strahler order for 5th correction etc.

[Fliesszeiten] #flow travel times
1                                   # 0 = ignore section, 1 = section is active
$path//$Grid//.fzt            # result grid flow travel times
$path//$Grid//.dep            # result grid river depths
$path//$Grid//.wit            # result grid river widths
$Manning                            # Manning coefficient (roughness
coefficient)
$SpecDisc                     # specific discharge filling the channel completey
$path//$Grid//.dhk       # input grid DEM
$path//$Grid//.slp       # input grid slope
$path//$Grid//.flk       # input grid flow direction (corrected from above)
$path//$Grid//.ack       # input grid flow accumulation (corrected from above)
$path//$Grid//.nuk       # input grid river network (corrected from above)

[Flusseparation] # separating rivers and computing river mouths
1                                   # 0 = ignore section, 1 = section is active
$path//$Grid//.m              # result grid with pour points (river mouths)
1                                   # smallest order for creating sub basins
6                                   # largest order for creating sub basins
$path//$Grid//.ork       # input grid Strahler orders (corrected from above)
$path//$Grid//.nuk       # input grid river codes (corrected from above)
$path//$Grid//.flk       # input grid flow direction (corrected from above)

[Einzugsgebiete] # defining subbasins
1                                   # 0 = ignore section, 1 = section is active
$Manning
$SpecDisc
$path//$Grid//.dhk            # input grid DEM (corrected)
$path//$Grid//.ack            # input grid flow accumulations (corrected)
$path//$Grid//.flk            # input grid flow directions (corrected)
$path//$Grid//.fzt            # input grid flow travel times
3                             # number of following blocks (each for one Order)
$path//$Grid//.m01            # pour points for order 1
$path//$Grid//.e01            # result grid sub basins for these pour points
$path//$Grid//.fs1            # result grid flow travel time sums
$path//$Grid//.ln1            # result grid routing channels (river links)
$path//$Grid//stru.e01 # routing structure to be used in WaSiM-control file
$path//$Grid//.m02            # pour points for order 2
$path//$Grid//.e02            # result grid sub basins for these pour points
$path//$Grid//.fs2            # result grid flow travel time sums
$path//$Grid//.ln2            # result grid routing channels (river links)
$path//$Grid//stru.e02  # routing structure to be used in WaSiM-control file
$path//$Grid//.pur            # pour point grid, externally generated
$path//$Grid//.ezg            # result grid sub basins for these pour points
$path//$Grid//.fzs            # result grid flow travel time sums
$path//$Grid//.lnk            # result grid routing channels (river links)
$path//$Grid//stru.ezg # routing structure to be used in WaSiM-control file
```

7.3 Error codes of WaSiM

return codes of WaSiM if the program is canceled by an error or by user break via Ctrl-C.

1 not enough memory (error allocating memory)

2 incompatible module combination (e.g. glacier and silting up or active tracer module when using multi-layer vegetation)

3 zone codes or region codes to small (<1) or to large (>32768)
4 too much zones (>16000)
5 error opening log-file
6 all cells are nodata in the flow travel time grid
7 all cells are nodata in the zone grid (or sub-catchment grid)
8 row index for point-statistic-files is larger than number of rows in the grid
9 column index for point-statistic-files is not in the range of valid values
10 error reading parameters from the control file (different modules)
11 error closing control file
12 not enough space for new grids in the "gridinfo"-array
13 number of levels in a stack read in from disk doesn't match the number of levels according to the soil table
14 error opening control file
15 error opening a statistics file
16 error closing a statistics file
17 parameter not found in control file
18 write error (writing binary grid)
19 read error (reading binary grids)
---- *20* *reserved*
21 error opening grid or stack
22 error reading grid or stack
23 error closing grid
24 error in grid measures (coordinates of lower left cell or number of rows/columns or cell size)
25 too big area with nodata-values within grid, which should not be nodata according to the zone grid
26 error creating grid or stack
27 error closing grid (like 22)
28 error opening control file
29 error in landuse table
30 error in soil table
31 wrong land use code or irrigation code (code from land use grid or irrigation grid not found in control file)
32 wrong soil type code (code from soil type grid not found in control file)
33 grid not found in memory (from a module during initialization)
34 paragraph not found in control file
35 error reading a routing description (parameter for reservoirs)
36 error comparing routing date with date from the external inflow file
37 error opening file for reservoir content or external abstraction
38 end of control file reached unexpected
39 error reading from external inflow file
40 string is too long (reading a routing description)
41 wrong subbasin code (different combinations possible for this error to occur)
42 other error in routing module
43 discharge not found in memory during initialization of the routing module
44 time step too short (e.g. for Hamon-evaporation time step must be larger than 1 day)
45 too less correction factors for Hamon-evaporation
46 data requirements not satisfied for evapotranspiration
47 infiltration module can only be started, if soil model is also active (only Topmodel version)
48 error reading scenario data for meteo-inputs
49 error opening/reading meteorological input data file

50 error closing met. input file
51 starting date not found in met. input file
52 wrong date in met. input file
53 wrong interpolation method given in control file
54 wrong interpolation parameters (anisotropy)
55 floating point error soil model (only Topmodel)
56 error reading tracer parameters
57 discharge not initialized
58 more than 9 tracers described in the control file
59 two ore more tracers have the same name (first 3 chars)
60 no routing is done, but irrigation is active (activate both, irrigation and routing)
61 error in flow travel time grid: the first time zone has no cells in one of the subbasins (this may happen, if the derivation of the travel flow time grid was based on an different subbasin grid than the one that is actually used for the modeling)
62 error opening statistic files for control point statistics
63 error in time step given in a sub-model compared to the time to interval end from the [output_interval] section
64 error opening storage content file
65 error opening statistics file for writing
66 error in cell size (cell size grid)
67 error in generating station mesh for bi-linear interpolation
68 not used
69 error in glacier model (deprecated, not used any longer)
70 error converting station coordinates into a grid (some interpolation methods)
71 error reading subbasin code, tracer code, reservoir code or tributary index from storage file
72 a parameter from the storage file is not equal to the one from the control file
73 a station or coordinate was not found in a meteorologic input file
74 No RelCelAreaGrid found (probably internal program error, send message to author, please)
--- *75-79 reserved*
80 glacier model error
81 memory allocation error for matrices
82 identical input coordinates for two stations in spline interpolation
83 Glacier ID grid contains a value where the glacierized fraction grid contains only 0 (fraction must by > 0 or the Glacier ID must be -9999)
84 glacier is located in more than one subbasin. Each single glacier may belong to one specific subbasin only
--- *85-90 reserved*
91 invalid method for solving the Richards-approach
--- *92-96 reserved*
97 wrong order of subbasin codes in unsatzon model (for sub-basin specific parameters)
98 program canceled by Ctrl-C (option: 2 no writing of buffers and grids)
99 program canceled by Ctrl-C (option: 1, writing all buffers and grids)
--- *100-109 reserved*
110 error in grain size distribution (used for silting-up model)
111 numbers of landuse layers has to be equal for all multilayer-landuse types
112 multiple landuse table: parameter Landuse_Layers not found
--- *113-119 reserved*
120 MPI-Regions: to much strips to be stored in a grid row in row
--- *121-130 reserved*
131 wrong soil table format – use new format as described in the documentation

--- 132 reserved

133 could not read "NumberOfEntities" keyword in RegionalSuperposition

134 error reading regions in RegionalSuperposition: the number of regions does not match the number of regions in the regions grid or the ascending order rule was broken

135 error reading weights for regions (in RegionalSuperposition)

--- 136-149 reserved

150 wrong Julian day number returned (internal error, please report to the author)

151 wrong Julian day fix-point (internal error, please report to the author)

152 dynamic phenology (method 1 or 2) or altitude correction resulted in an invalid Julian day

153 wrong Julian day number returned (internal error, please report to the author)

154 wrong month returned for a given Julian day

155 wrong month returned for a given Julian day

--- 156-198 reserved

198 feature “periodicity” in [standard_grids]: invalid combination of additional parameters

199 river link from the river link grid is not valid for the current routing structure -> there is no routing description for a river with this code

--- 200 reserved

201 file buffer for input files (e.g. external inflows) cannot be read correctly (file corrupt?)

202 file buffer pointer for input files cannot be positioned correctly (file corrupt?)

203 Error in localizing the actual date in the discharge input file (for comparing modelled with observed discharges)

211 error in initialization of evapotranspiration model when initializing lake model

212 error in initialization of soil-model when initializing lake model

213 error in initialization of discharge routing model when initializing lake model

214 error in initialization of groundwater model when initializing lake model

215 error in initialization of unsatzon-model when initializing lake model

223 reserved

224 lake must be located completely within one sub-basin

--- 225-299 reserved

300 Error in connecting internal abstractions with internal inflow – inflow is already connected to an internal abstraction. Each abstraction (AL-element) must be connected to exactly one internal inflow (ZL-element)

301 Error in connecting internal abstractions with internal inflows: target and source of an internal abstraction are not allowed to point to the same sub-basin

302 Error in connecting internal abstractions with internal inflows: internal inflow has no connection to an internal abstraction (no abstraction with that code found)

303 Error in connecting internal abstractions with internal inflows: internal abstraction has no connection to an internal inflow (no inflow with that code found)

--- 304-310 reserved

311 land use code in land use grid is not present in the multilayer-landuse-table in the control file

312 not all layer codes from a land use from the multi-layer land use table are present in the landuse-table in the control file (add this type in the landuse table)

313 the first Layer code of each MultiLayer-Landuse entry in the maulti layer land use table must be a valid entry (not 9999 or larger)

--- 314-319 reserved

320 Error opening diagnostics output file (XML-diag file)

--- 321-399 reserved

400 error looking for opening bracket “{“ after "SiltingUpExpressions" (when defining user-equations for silting-up)

401 error looking for closing bracket “}” after "SiltingUpExpressions" (when defining user-

equations for silting-up)

402 no SiltingUpExpressios found (at least X, Y and Z must be initialized)

403 expression could not be parsed correctly (in silting-up model)

404 more parameters are expected in snow model for methods 5 to 11

--- 404-1000 reserved

1001 error deleting semaphore file (coupling with external models – reading and writing grids)

1002 error deleting external grid (coupling with external models – reading and writing grids)

1003 error looking for internal grid to read from an external source with given internal name. Possible reason: internal name of the grid is wrong (case sensitive!)

1004 error looking for internal grid to write to an external target with given internal name. Possible reason: internal name of the grid is wrong (case sensitive!)

1005 error looking for zonal statistic with a given internal name. Possible reason: name of the statistic entity is wrong (case sensitive!)

1006 error in defining special outputs: count of X coordinates dos not match the number of Y coordinates.

1007 error opening output file for special outputs.

1008 location of a cell for special outputs is outside the actual domain (check X and Y coordinates)

1009 configured external coupling when running in MPI mode with more than one instance (use non-MPI version for coupling with external programs)

1030 Grid wit Hydrologic Respond Units not found internally although the parameter in section [SpinUp] defined the grid with HydrologicRespondUnits to be found internally (as standard grid)

1035 temperature interpolation not found. This is required for pre-SpinUp! (no regions should be used for pre-SpinUp)

1040 heat transfer model not active in unsatzon_model. This is required for pre-SpinUp!

1041 heat transfer model not active in unsatzon_model. This is required for Layered Snow Model!

1050 interpolation method 9 (read from grid) no valid grid name was found (-9999 as input) and no last valid grid name present (i.e., simulation starts, when input list file contains -9999 only)

1060 dumpfile has been written and command line option -closeAfterDump was active, so the model aborts now

1061 wrong argument passed to the model as max number of dumpfiles to write

1100 number of netCDF files that can be open at the same time in a sub model exceeded (default is 200)

1150 MPI with re-infiltration does not work with tracers yet. Please use Version prior to 10.03.00

1160 Snow Interception and Canopy Energy Balance doesn't work with simple snow melt approaches. Use an energy balance approach instead

1161 method is not supported in interception model (use one of 1, 2, 11, 12, 21, 22, 31, 32). See sample control file for the meanings of each method

1200 error opening data assimilation listfile (e.g. with SWE-gridnames)

1300 error reading start or end date and time in section `[output_interval]`

1301 error reading start or end date and time in section `[output_interval]` when using new keywords

1302 interpolation methods 2 and 3 with externally prepared regression files does not work with time steps less than one hour (starting with version 10.05.00). Use interpolation methods 10 and 11 instead with internal regression

1400 routing mode_ error opening the input file with total runoffs for all subbasins when WaSiM runs with option -mode:routing. Make sure to provide the file which is written in a former model run as total runoff data from [unsatzon_model] or [surface_routing]

1410 routing mode: number of expected columns in the external input file with total subbasin runoff values doesn't match the number of subbasins defined in the control file

1420 routing mode: error looking for starting date in the external input file with total subbasin runoff values

1501 ArcVeg integration is only available if heat_transfer is switched on

1502 ArcVeg climate zpn count (AVNumClimZon) and PGP count (AVNumPGPs) must be both provided and be >= 0

1503 ArcVeg AVNMinParam for climate zone <climate zone> and PGP <PGP> could not be read in correctly

1504 ArvVeg actual climate zone not in range of NMin params dimensions as read in before

1505 ArcVeg actual PGP <PGP> count is larger than allowed in the control file (AVNumPGPs)

1506 ArvVeg land use type to be used for thickness change (AVLUSEType) not in range (a valid index of the land use table must be used)

1507 ArcVeg Layer count of soil layers to be changed in thickness (AVLayerCnt) not in range (either 0 or more layers than the soil table defines)

1508 ArcVeg dry density of soil layers to be changed in thickness (AVSoilDryDensity) not in a feasible range of 1 to 1000 Kg/m^3

1600 sum of soil grain fractions for clay, silt and sand is zero, no calculation of soil type possible

1601 depths of soil table entries must be at least differ by factor 1.2 between shallow, medium and thick soils. Also, the first three entries of the soil table must be a shallow, a medium and a thick soil, in this order. The actual soil table doesn't fullfill this requirement.

7.4 Internal names for grids that can be used for writing special output

The following identification strings can be used in the new section [special_output] to define entities. See also the very end of the control file example in appendix 1. More documentation is in preparation. Comments are added where it may not be clear what is meant by the name of the identity

```
global_radiation
net_radiation
sunshine_duration
temperature
temperature_14
wind_speed
vapor_pressure
vapor_pressure_14
air_humidity
air_humidity_14
air_pressure
precipitation
available_soil_moisture  (Topmodel version only)
albedo
soil_storage             (Topmodel version only)
ETR2ETP                  ratio of real to potential evaporation
fillable_porosity        (Topmodel version only)
saturation_deficite      (Topmodel version only)
throughfall
snowcover_outflow        melt plus liquid precipitation
melt_from_snowcover      pure melt
days_snow
snow_age
```

```
snow_fraction             solid precipitation
rain_fraction             liquid precipitation
melt_from_firn
melt_from_ice
unsaturated_zone_storage (Topmodel version only)
interflow_storage         (Topmodel version only)
saturated_soil_partition (Topmodel version only)
soil_storage_old          (Topmodel version only)
baseflow                  baseflow/exfiltration into rivers QB
directflow                overland runoff QD
interflow                 interflow QI
peakflow                  (Topmodel version only)
temperature_night
corr_temperature
pot_transpiration         potential evapotranspiration
real_transpiration
potential_evaporation     potential evaporation from soil
real_evaporation
snow_evaporation
potential_InterceptionEvap potential interception evaporation
real_InterceptionEvap
Shadow                    (yes = 1, no = 0)
Ieff_max                  maximum theoretical incoming shortwave
                          radiation
infiltration_excess
total_infiltration
interception_storage
interception_evaporation
snowstorage_solid         valid for unglacierized fractions only
snowstorage_liquid        valid for unglacierized fractions only
root_depth
vegetation_coverage_degree
leaf_area_index
theta                     water content of the soil (stack)
hydraulic_heads
geodetic_heads
q_down                    flow between soil layers (stack)
groundwater_distance      groundwater table below surface
canopyresistance1
Theta_in_GWLEVEL
groundwater_recharge
groundwater_level         index of the soil layer
drainage_flow
ponding_storage
Max_Pumping
SB_1_grid                 soil moisture relative 0...rot depth
SB_2_grid                 soil moisture absolute in mm
ROOTgrid
river_infiltration        re-infiltration into groundwater
river_exfiltration        = base flow
groundwater_balance
macropore_infiltration
gw_head_                  groundwater head
```

```
gw_flow_x_                flow velocity x direction
gw_flow_y_                flow velocity y direction
gw_leakage_
gw_balance
snow_evaporation
SurfaceCanopyResistance
SurfaceIntercepResistance
SurfaceEvaporationResistance
RoughnessLength
Capillar_Uprise
Percolation
snow_storage_total
AspectCorrectedGlobalRadiation
SumOfForcingUnits
Pheno_start
SumOfChillingUnits
Fstar_ForcingThreshold
MultipleStartTimes        (stack)
MeanThetaInRootZone
SiltingUpDisposition
RainPause
KineticEnergy
TemperatureAVG
TemperatureMOVAVG7
SnowFreeDaysGrid
SnowCoverDaysGrid
PermafrostThawDepth       when using simple permafrost model
firn_WE_stack             (stack with firn layers, e.g. 7 layers)
GlacierMassBalance
OldGlacierMassBalance
GlacierizedCells
debris_on_glaciers
Soil_Temperature_Stack    (stack with soil temperatures)
Soil_SE_Stack             stack with relative saturation values
                          (i.e. water/ice relation within each soil
                          layer)
ThawDepthTMod             thaw depth when using heat transfer
T_Lower_Boundary_Condition (used for constant temperature AND heat
                           flux boundary condition initialization)
HeatFlux_Lower_Boundary_Condition (used as constant heat flux
                           lower boundary condition)

frozenSoilInGroundwater

TotalSnowStorage_onGlacier    valid for the glacierized fraction
SolidSnowStorage_onGlacier    valid for the glacierized fraction
LiquidSnowStorage_onGlacier   valid for the glacierized fraction
SnowMelt_onGlacier            valid for the glacierized fraction
SnowEvaporation_onGlacier     valid for the glacierized fraction

→ additional definitions for snow model extensions (Warscher,
2011-2013)
snowtemp                  temperature of snow pack (unglac. part)
```

surftemp	temperature of snow surface (unglac.)
snowtemp_onGlacier	temperature of snow pack, glacierized fraction of a cell only
surftemp_onGlacier	temperature of snow surface, glacierized fraction of a cell only
ebal_snow	energy balance for snow melt approaches
radflux	net radiation flux (new energy balance snow melt appr.)
sensflux	sensible heat flux (new energy balance snow melt appr.)
latflux	latent heat flux (new energy balance snow melt appr.)
adflux	advective heat flux (new energy balance snow melt appr.)
soilflux	soil heat flux (new energy balance snow melt appr.)
input_mass	input to snow slides
mobile_mass	mobile mass (in a snow slide)
deposition	deposition mass of a snow slide

HydrologicRespondUnits

scalinggrid_<meteoentity>	if read in as standard grid, e.g. as scalinggrid_precipitation, it will be used after interpolation to scale the cell values by multiplying each cell of the interpolation result with the matching cell value of the scaling grid.
SurfaceRoutingFluxes	Stack, containing flow on surface (surface routing) in m^3/s. Level 0 contains the sum of all 8 directions (only positive entries, so it may not what you will expect when looking for a balance). Level 1 to 8 contain the fluxes for flow directions 1 (N), 2 (NE), 3 (E), 4 (SE), 5 (S), 6 (SW), 7 (W) and 8 (NW). Level 10 is empty.

SurfStorSiltingUp

SurfaceRoutingHead