Introduction.

This help file describes the enhanced modules written for application in CRHM_new.

Using Demonstration Models.

Complete sample projects are distributed with CRHM. These may be found in the directory "CRHM_project_examples" located in the same directory as the CHRM executable.

The subdirectories "Bad Lake", "Marmot Creek", "Smith Creek" and "Yukon" contain complete example projects for different Canadian regions. These projects may be executed from CRHM using the File/Open menu or double clicking the project file from Windows Explorer.

Within CRHM examples of models using different combinations of modules can be constructed using the pull down menus in the Build menu screen. To use them use the following steps.

- 1. Run CRHM.
- 2. Goto the Build/Construct screen to select the desired number of HRUs and exit this screen using Build to register the selected number of HRUs..
- 3. In the Build screen select the example model desired from the list.
- 4. In the Observation screen select a suitable observation file. The file "C:\Program Files\CRHM\Examples\Winter1974.obs" can be used for demonstration purposes only.
- 5. Select some variables to display and click on Run to execute..

This is a training exercise. The usual warnings apply, i.e. the selection of modules, parameters and set of observations must be relevant to the physical application to achieve meaningful results.

Notes.

- It is best to select the number of HRUs before selecting a sample model, then the default parameters are optimum. When the number of HRUs is increased after selecting the model, the last HRU parameter value in the earlier project is duplicated for the new HRUs.
- If a project is already loaded when a sample project is selected, the current modules loaded will be replaced by the model selected. The observation files and number of HRUs will not be affected.
- When a sample model is selected, its description is added to the first line of

the prospective project file and also inserted into the TChart title.

• The user must customise all parameters in the model to define their application.

Module Variations.

The original CRHM module is very limited in its loading ability and could not be made to load in different modes without a lot of difficulty and unused portions of the module still visible. The input variable/observations, output/observations variables and required parameters were all fixed and always visible. An attempt was made to work around this by making observations optional and modules generating declared observations and the use of the AKA screen. This allowed modules to perform in different ways as dictated by the available driving data. However, this was not very easy for the user to use. There were also numerous unused parameters and variables as these had to be declared for all possible uses of the module and the user had to know which ones applied to the current module application.

Variations were introduced to overcome the limitations outlined above. Module names have been extended as follows. The original name say "longVt" will run as always. However, variations of the original module will be possible, denoted as "longVt#1", "longVt#2", etc. These variations are enhanced versions of the original module, with different inputs requirements, say synthetic shortwave "QsiD_Var" (Annandale) instead of the measured observation Qsi. The parameters and input/output variables can also be customised. The module variations can also have different algorithms. Descriptions of current variation modules follow.

Annandale - short-wave estimation.

- 1. Annandale outputs QsiA#, QsiD# and QsiS# short-wave observations (interval, daily and interval on slope) and hru_SunAct variable.
- 2. Annandale#1 outputs QsiA_Var, QsiD_Var, QsiS_Var (interval, daily and interval on slope) variables with hru_SunAct variable.

calcsun - sunshine hours.

- 1. calcsun input SunAct Obs or Qsi Obs to calculate Qsitot if sunshine hours not available.
- 2. calcsun #1 Qsi observation to calculate Qsitot.
- 3. calcsun#2 QsiDobs (W/m^2) observation.

4. calcsun#3 - QsiD_Var (W/m^2) from Annandale.

Canopy, CanopyClearing and CanopyClearingGap - calculate short, long and all-wave radiation components at the snow surface.

- 1. CanopyClearing input Qsi and Qli observations.
- 2. CanopyClearing#1 input Qsi observation and QliVt_Var from longVtX etc..
- 3. CanopyClearing#2 input Qli observation and Qsi_Var from longVtX etc..
- 4. CanopyClearing#3 input Qsi_Var and QliVt_Var from longVtX etc..

ebsm - energy-budget snowmelt model for the Canadian Prairies.

- 1. ebsm no inputs for Brunt. Optional temperature and radiation index. QnD can be used with Use_QnD for legacy projects.
- 2. ebsm#1 input interval variable Qnsn_Var (W/m^2) from CanopyClearing etc..
- 3. ebsm#2 input interval observation Qnsn (W/m^2).
- 4. ebsm#3 input daily observation QnD (MJ/m^2*d).

longVt - long-wave using terrain view factor.

- 1. longVt module input Qsi observation.
- 2. longVt#1 module input QsiDObs observation from module Slope_Qsi.
- 3. longVt#2 module input QsiD_Var observation from module AnnandaleX#1 etc..

Needleleaf - calculate short, long and all-wave radiation components at the snow surface.

- 1. Needleleaf input Qsi and Qli observations.
- 2. Needleleaf#1 input Qsi observation and QliVt_Var from longVtX etc..
- 3. Needleleaf#2 input Qli observation and Qsi_Var from longVtX etc..
- 4. Needleleaf#3 input Qsi_Var and QliVt_Var from longVtX etc..

obs - handle driving observations.

- 1. obs standard version to handle interval observations.
- 2. obs#1 inputs daily t_max and t_min to generate hru_tmax and hru_tmin for Annandale.

pbsm - blowing snow transport.

- 1. pbsm input hru_u.
- 2. pbsm#1 input hru_Uadjust from walmsley_wind etc.

pbsmSnobal - blowing snow transport.

- 1. pbsmSnobal input hru_u.
- 2. pbsmSnobal#1 input hru_Uadjust from walmsley_wind etc.

Soil - soil moisture throughout the year.

- 1. Soil standard version.
- 2. Soil#1 runoff is limited by culvert flow.

Slope_Qsi - modified short-wave on a slope.

- 1. Slope_Qsi output declared observations QsiS# and QsiDObs#.
- 2. Slope_Qsi#1 output variables QsiS_Var and QsiD_Var.

SnobalCRHM - energy balance snowmelt model.

- 1. SnobalCRHM input Qsi and Qli observations.
- 2. SnobalCRHM#1 input variables Qsisn_Var and Qlisn_Var from module CanopyClearing.
- 3. SnobalCRHM#2 input observation Qsi and QliVt_Var from module longVtX etc..
- 4. SnobalCRHM#3 input variables QsiS_Var from Annandale and QliVt_Var from module longVtXetc..

REW_route - handles the routing of surface and subsurface runoff from RBs (representative basins).

1. REW_route - uses Muskingum for routing.

2. REW_route#1 - uses Clark for routing.

How to use Module Variations.

The variation modules behave like any other module. The only difference is that in the construct screen there are additional steps to selecting them. In the construct screen the modules available are displayed in the listbox to the left of the screen. The modules displayed are determined by the level(s) selected in the "Module Level" list. When a module is selected in the modules available listbox the description of the module is displayed in the information area in the bottom left hand side of the screen. When the module has variations additional lines are displayed giving a list of the possible variations. Add the module to your project as normal. Now, when the module is right clicked in the Modules Selected listbox two additional options are available, Next variation and Last variation. Use these selections to select the variation you want. The requirements and outputs of the module will automatically be displayed in the adjacent module description area. At this point proceed as normal using the Check/Build/Cancel options to develop your project.

obs

Defined in Classobs. This module reads the climatic data from the observation data file(s) into the model. The time step is determined by the observation file interval. For normal use, the interval will be half or one hour. The module copies the observations to the HRU's with the requested corrections for redirection, height and climate change. One of the precipitation observations "p" and "ppt" must be present. If both are used (e.g. "p" for rainfall and "ppt" for snowfall) they are merged to generate "hru_p". When both "p" and "ppt" are used together, both must be present in every interval and set to zero when not used. "ppt" can also be in a separate "daily" file containing the daily rainfall. When "ppt" is combined with the interval data only the first interval value of every day is read and used as the total precipitation for the day. All other values are ignored.

Observations

- u (m/s) average wind speed over time step.
- p (mm/int) interval precipitation. Optional if ppt available. p and ppt values are merged.
- ppt (mm/d) daily precipitation. Optional if p available. p and ppt values are merged.
- t_max (°C) variation #1 daily maximum temperature. Only used for daily 24 hour time step.
- t_min (°C) variation #1 daily minimum temperature. Only used for daily 24 hour time step.
- t (°C) air temperature using the INTVL declobs function to read the entire days temperature data. Then modified in buffer by parameters lapse rate and ClimChng_t.
- rh (%) relative humidity using the INTVL declobs function to read the entire days relative humidity data.
- ea (kPa) vapour pressure using the INTVL declobs function to read the entire days vapour pressure data.

N.B. Either rh or ea are used. If both are available only rh is used. The other is calculated from the array of rh or ea values, using the adjusted temperature (lapse rate and climate change) and the parameter ElevChng_flag (maintain rh or ea with temperatue change).

Variables

The following are from interval observations.

- tday_intvls (°C) temperature array holding interval values. Values have Dimensioned [interval][dim].
- rhday_intvls (%) -RH array holding interval values. Dimensioned [interval][dim].
- eaday_intvls (kPa) ea array holding interval values. Values are calculated from above t and RH. Dimensioned [interval][dim].
- t_obs (°C) temperature array holding the original observation interval values before adjustment for lapse rate and climate warming. Note that HRU_OBS redirection have been applied. Access t-obs [period in day, dim].

The following variables are derived for each HRU from the observations.

- hru_t (°C) temperature
- hru_rh (%) relative humidity.
- hru_ea (kPa) vapour pressure
- hru_u (m/s) wind speed.
- hru_p (mm/int) precipitation. Calculated from the interval precipitation (p) and/or (daily precipitation (ppt) /frequency) or all the daily precipitation (ppt) during the first daily interval of the day as determined by the parameter "ppt_daily_distrib".
- hru_rain (mm/int) rain.
- cumhru_rain (mm) cumulative rain
- hru_snow (mm/int) snow.
- cumhru_snow (mm) cumulative snow.
- cumhru_snow_meas (mm) cumulative snow before catch adjustment.
- hru_newsnow () 0/1 for no/yes in the last interval.
- hru_tmean (°C) daily mean temperature.
- hru_tmax (°C) daily maximum temperature.
- hru_tmin (°C) daily minimum temperature.
- hru_umean (m/s) daily mean wind speed.
- hru_rhmean (%) daily mean RH.
- hru_eamean (kPa) daily mean vapour pressure.
- Pa(kPa) atmospheric pressure.

Parameters

- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- hru_elev (m) altitude.
- obs_elev (m) altitude.
- HRU_OBS () array[5][dim] of values indexing observations to HRUs. The order is 1) t, rh and ea; 2) p and ppt; 3) u; 4) Q; and 5) for special use.
- snow_rain_determination() snow/rain determination: 0 air temperature, 1
 ice bulb temperature.
- tmax_allrain (°C) precipitation is all rain when the temperature is greater or equal to this value.
- tmax_allsnow (°C) precipitation is all snow when the temperature is less or equal to this value.
- catchadjust () none 0/Nipher 1/MacDonald-Alter (not recommended) 2/Smith-Alter 3.
- ppt_daily_distrib () 0/1 daily precip all in first interval /equally divided over the day.
- lapse_rate (°C/100m) lapse rate correction.
- ElevChng_flag () Elevation change control; 0 maintain RH, 1 keep Vp within Vsat maximum.
- precip_elev_adj () precipitation height adjustment {adjusted p(or ppt) = p(or ppt)*(1.0 + precip_elev_adj*elev_difference/100)}.
- ClimChng_t (°C) Climate change additive temperature change.
- ClimChng_flag () Climate change control; 0 maintain RH, 1 keep Vp within Vsat maximum
- ClimChng_precip () Climate change multiplative p/ppt change {adjusted p(or ppt) = p(or ppt)*ClimChng_precip}.
- Use_Observations_As_Supplied () 0 -> CRHM handles observation corrections for lapse rate/allocation etc., 1 -> Externally corrected observation dataset.

Variable Inputs

• none

Notes

The parameter HRO_OBS applies to all observations. E.g t, rh, u, ea, Qsi, Qli, Tg etc..

The parameter HRO_OBS is normally defined in the module "obs".

The parameter HRO_OBS has a default value of 1 for every HRU. This implies that there is only one set of observations and that all HRUs use this set of values. However, if the observation file had six sets of temperature data, the observation has "t 6 (°C)" in its header then the following HRU_OBS settings are possible.

HRU_OBS = 1, 2, 3, 4, 5, 6 then the HRU1, HRU2 ... use temperatures T1, T2, T3 ... as driving inputs.

HRU_OBS = 4, 5, 6,1, 2, 3 then the HRU1, HRU2 ... use temperatures T4, T5, T6, T1, T2, T3 as driving inputs.

HRU_OBS = 4, 4, 4, 4, 4, 4 then the HRU1, HRU2 ... use only temperatures T4 as the driving input.

However, if the observation file has only three sets of temperature data, i.e. "t 3 (°C)" in its header, the last defined observation is duplicated for the higher values.

HRU_OBS = 4, 5, 6,1, 2, 3 then the HRU1, HRU2 ... use temperatures T3, T3, T3, T1, T2, T3 as driving inputs.

When the observation variables are displayed in CRHM they are displayed in the same order as in the observation file. When the observations are displayed from the HRU, e.g. hru_t, the order will be determined by the HRU_OBS values.

Basin

Defined in Classbasin. This module contains no active code. It declares general parameters for the model. Examples of the parameters declared are: basin area, <u>HRU</u> area, latitude, elevation, ground slope (GSL) and aspect angle (ASL).

Observations

• none

Variables

• run_ID () - run identification.

Parameters

- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- hru_lat (°) latitude.
- hru_elev (m) altitude.
- hru_GSL (°) ground slope.
- hru_ASL (°) aspect.
- basin_name text string.
- hru_names text strings.
- RUN_ID integer number used to identify CRHM model runs. When it is positive it is appended to the log file name "CRHM_output".
- RUN_START run start time (Automation).
- RUN_END run end time (Automation).
- INIT_STATE initial state file (Automation).
- RapidAdvance_to element[1] date to advance to mm/dd/yyyy.
- Loop_to element[1] date to loop to mm/dd/yyyy and element[2] number of loops before continuing.
- StatVars_to_Handle state variables to allow to change.

Variable Inputs

• none

Notes

- 1. If the sum of the HRU areas is not within 0.01 km² of the basin area, the basin_area is set to the sum of the HRU areas.
- 2. The parameter values specified in basin override any values specified in the model modules. The values in the modules are the default values for testing.
- 3. The basin parameters are overriden when the parameters are edited and saved in a project or parameter file.
- 4. If basin_name or hru_names contain embedded spaces the individual names nust be enclosed in single quotes.
- 5. The first field defined in the parameter 'StateVars_to_Handle' has special significance. If it is explicit, i.e. 'soil_moist@A' then all other fields are matched exactly. However, if the first parameter is a root CRHM variable name, e.g. 'soil_moist' it is treated as a wildcard and will match any group having the 'soil_moist' variable, i.e. 'soil_moist@A', 'soil_moist@B', 'soil_moist@C' etc..

NO_pbsm.

Defined in ClasspNO_bsm. This module calculates snow water equivalent from snow fall. Sublimation will be added later. It is intended to be used instead of the Prairie Blowing Snow Model when there is no blowing snow. An example would be in a forest.

Observations

• none

Variables

- SWE (mm) snow water equivalent. State variable.
- cumSno (mm) snow (net_snow) accumulation from beginning of winter.

Parameters

- basin_area (km2) basin area.
- hru_area (km2) hru_area.
- inhibit_evap (flag) an output parameter set true when the SWE is greater than zero. It is used to inhibit evaporation from the evaporation modules.

Variable Inputs

- newsnow () 0/1 for no/yes from module obs.
- net_snow (mm Δt) snow fall from wild module intcp, brushintcp etc.

Notes

pbsm (Pomeroy and Li, 1999)

Defined in Classpbsm. This module calculates snow transport and sublimation. The snowfall observation can be a daily total or the actual interval snowfall. The former case is handled by assuming that the snowfall is uniformly distributed over every interval of the day. The transport and sublimation of blowing snow are calculated every interval using the interval wind speed, air temperature and relative humidity. The model has been extended to handle the transport of snow between HRUs. At the end of a day when snow transport has occurred, snow transported from HRUs with low roughness is distributed over the HRUs with greater roughness according to the fractions specified in the distribution parameter. No transported snow enters the HRU with the lowest roughness. When HRUs with lower vegetative height fill to their maximum, the excess is distributed over the remaining unfilled HRUs.

Observations

• none

Variables

- SWE (mm) snow water equivalent. State variable.
- Subl (mm Δt) the mass of snow lost from an HRU by sublimation in time step, Δt expressed as an equivalent average depth of water over an HRU.
- cumSubl (mm) cumulative sublimation.
- Drift (mm ∆t) the mass of snow lost from an HRU by snow transport in time step, ∆t – expressed as an equivalent average depth of water over an HRU.
- cumDrift (mm) cumulative transport
- cumSno (mm) snow (net_snow) accumulation from beginning of winter.
- Prob () interval probability of blowing snow.
- BasinSnowLoss (mm Δt) transport out of basin.
- cumBasinSnowLoss (mm) cumulative transport out of basin.
- cumBasinSnowGain (mm) cumulative transport into basin.
- snowdepth (m) calculated snow depth from SWE (Gray/Pomeroy).

Parameters

- fetch (m) fetch distance.
- Ht (m) crop height.
- distrib () distribution fractions. Value for HRU 1 controls snow transport into the basin.
- N_S (1/m^2) vegetation number density.
- A_S (m) stalk diameter or silhouette.
- basin_area (km2) basin area.
- hru_area (km2) hru_area.
- inhibit_evap (flag) an output parameter set true when the SWE is greater than zero. It is used to inhibit evaporation from the evaporation modules.
- inhibit_bs (flag) an input inhibiting blowing snow when set equal to 1. Inhibited HRU is still able to receive drift from other HRUs.

Variable Inputs

- hru_t (°C) air temperature from module obs.
- hru_rh (%) relative humidity from module obs.
- hru_u (m/s) wind speed from module obs.
- newsnow () 0/1 for no/yes from module obs.
- net_snow (mm Δt) snow fall from wild module intcp, brushintcp etc.

Notes

- 1. The parameter *fetch* cannot be less than 300m.
- 2. The parameter *distrib* for the first HRU is used differently from all the other HRUs. It determines the drift into the model basin based upon the drift out of the first HRU. BasinSnowGain is equal to distrib[HRU1]*Drift[HRU1].
- 3. The first HRU, i.e. the lowest vegetation height. When the snow is redistributed, the *distrib* values for the HRU's not filled to their vegetation height are summed and each HRU receives its share. The sum of distributions need not add to 1.0.

E.g. if the transport D is redistributed over HRUs: A, B and C with their *distrib* parameter having values of a, b and c respectively. The snow transport would be distributed as:

a*D/(a + b + c), b*D/(a + b + c) and c*D/(a + b + c). Note that any excess from a HRU filling to its vegetation height is deposited in the last HRU, i.e. the one having the tallest vegetation.

3. If the value of *distrib* is less than 0 for an HRU, all snow transport to its capacity is assumed to be caught by this HRU. The next HRU is the beginning of a new blowing snow regime.

4. Cumulative drift from the last HRU is handled as follows:

distrib > 0.0 - the drift is handled as described above. Any excess after the vegetation is full is transported out of the basin - i.e. added to 'BasinSnowLoss'.

distrib < 0.0 - all drift is deposited in the hru.

distrib == 0.0 - all drift is transported out of the basin - i.e. added to 'BasinSnowLoss'.

pbsmSnobal (Pomeroy and Li, 1999)

Defined in Classpbsm. This module calculates snow transport and sublimation. The snowfall observation can be a daily total or the actual interval snowfall. The former case is handled by assuming that the snowfall is uniformly distributed over every interval of the day. The transport and sublimation of blowing snow are calculated every interval using the interval wind speed, air temperature and relative humidity. The model has been extended to handle the transport of snow between HRUs. At the end of a day when snow transport has occurred, snow transported from HRUs with low roughness is distributed over the HRUs with greater roughness according to the fractions specified in the distribution parameter. No transported snow enters the HRU with the lowest roughness. When HRUs with lower vegetative height fill to their maximum, the excess is distributed over the remaining unfilled HRUs. This is a modification of the original PBSM. SWE is now a state variable of Snobal accessed by pbsmSnobal using a PUT.

Observations

• none

Variables

- hru_subl (mm Δt) the mass of snow lost from an HRU by sublimation in time step, Δt expressed as an equivalent average depth of water over an HRU.
- cumSubl (mm) cumulative sublimation.
- hru_drift (mm Δt) the mass of snow lost from an HRU by snow transport in time step, Δt – expressed as an equivalent average depth of water over an HRU.
- Drift_out (mm Δt) the mass of snow lost from an HRU by snow transport in time step, Δt expressed as an equivalent
- Drift_in (mm Δt) the mass of snow gained from an HRU by snow transport in time step, Δt expressed as an equivalent
- cumDrift (mm) cumulative transport.
- cumSno (mm) snow (net_snow) accumulation from beginning of winter.
- Prob () interval probability of blowing snow.

- BasinSnowLoss (mm Δt) transport out of basin.
- cumBasinSnowLoss (mm) cumulative transport out of basin.
- cumBasinSnowGain (mm) cumulative transport into basin.
- snowdepth (m) calculated snow depth from SWE (Gray/Pomeroy).

Parameters

- fetch (m) fetch distance.
- Ht (m) crop height.
- distrib () distribution fractions. Value for HRU 1 controls snow transport into the basin.
- N_S (1/m^2) vegetation number density.
- A_S (m) stalk diameter or silhouette.
- basin_area (km2) basin area.
- hru_area (km2) hru_area.
- inhibit_evap (flag) an output parameter set true when the SWE is greater than zero. It is used to inhibit evaporation from the evaporation modules.
- inhibit_bs (flag) an input inhibiting blowing snow when set equal to 1. Inhibited HRU is still able to receive drift from other HRUs.

Variable Inputs

- hru_t (°C) air temperature from module obs.
- hru_ea (kPa) vapour pressure from module obs.
- hru_u (m/s) wind speed from module obs.
- net_snow (mm Δt) snow fall from wild module intcp, brushintcp etc.
- SWE (mm) snow water equivalent. Put.
- z_s (m) snowcover depth. Put.
- rho (kg/m^3) snow density. Put.

Notes

- 1. The parameter *fetch* cannot be less than 300m.
- 2. The parameter *distrib* for the first HRU is used differently from all the other HRUs. It determines the drift into the model basin based upon the drift out of the first HRU. BasinSnowGain is equal to distrib[HRU1]*Drift[HRU1].
- 3. The first HRU, i.e. the lowest vegetation height. When the snow is redistributed, the *distrib* values for the HRU's not filled to their vegetation

height are summed and each HRU receives its share. The sum of distributions need not add to 1.0.

E.g. if the transport D is redistributed over HRUs: A, B and C with their *distrib* parameter having values of a, b and c respectively. The snow transport would be distributed as:

a*D/(a + b + c), b*D/(a + b + c) and c*D/(a + b + c). Note that any excess from a HRU filling to its vegetation height is deposited in the last HRU, i.e. the one having the tallest vegetation.

3. If the value of *distrib* is less than 0 for an HRU, all snow transport to its capacity is assumed to be caught by this HRU. The next HRU is the beginning of a new blowing snow regime

4. Cumulative drift from the last HRU is handled as follows:

distrib > 0.0 - the drift is handled as described above. Any excess after the vegetation is full is transported out of the basin - i.e. added to 'BasinSnowLoss'.

distrib < 0.0 - all drift is deposited in the hru.

distrib == 0.0 - all driftt is transported out of the basin - i.e. added to 'BasinSnowLoss'.

sbsm (Richard Essery, Long Li and John Pomeroy, 1999)

Defined in Classsbsm. This module is a simplified blowing snow model which can reproduce PBSM results very closely with much less computational effort. The snowfall observation can be a daily total or the actual interval snowfall. The former case is handled by assuming that the snowfall is uniformly distributed over every interval of the day. The transport and sublimation of blowing snow are calculated every interval using the interval wind speed, air temperature and relative humidity. The model has been extended to handle the transport of snow between HRUs. At the end of a day when snow transport has occurred, snow transported from HRUs with low roughness is distributed over the HRUs with greater roughness according to the fractions specified in the distribution parameter. No transported snow enters the HRU with the lowest roughness. When HRUs with lower vegetative height fill to their maximum, the excess is distributed over the remaining unfilled HRUs.

Observations

• none

Variables

- SWE (mm) snow water equivalent. State variable.
- wet_snow (mm) wet snow mass
- Subl (mm Δt) the mass of snow lost from an HRU by sublimation in time step, Δt expressed as an equivalent average depth of water over an HRU.
- cumSubl (mm/day) daily sublimation.
- Drift (mm Δt) the mass of snow lost from an HRU by snow transport in time step, Δt expressed as an equivalent average depth of water over an HRU.
- cumDrift (mm/day) daily transport.
- cumDriftIn (mm/day) cumulative transport to HRU. (local variable).
- cumSno (mm) snow (net_snow) accumulation from beginning of winter.
- Prob () interval probability of blowing snow.
- snow_age (hr) snow age. (local variable).
- BasinSnowLoss (mm Δt) transport out of basin.
- cumBasinSnowLoss (mm) cumulative transport out of basin.

• cumBasinSnowGain (mm) - cumulative transport into basin.

Parameters

- fetch (m) fetch distance.
- Ht (m) crop height.
- Zr (m) Ratio of aerodynamic roughness length to vegetation height. See below for typical values.
- distrib () distribution fractions. Value for HRU 1 controls snow transport into the basin.
- basin_area (km2) basin area.
- hru_area (km2) hru_area.
- inhibit_evap (flag) an output parameter set true when the SWE is greater than zero. It is used to inhibit evaporation from the evaporation modules.

Variable Inputs

- hru_t (°C) air temperature from module obs.
- hru_rh (%) relative humidity from module obs.
- hru_u (m/s) wind speed from module obs.
- newsnow () 0/1 for no/yes from module obs.
- net_snow (mm Δt) snow fall from wild module intcp, brushintcp etc.

Notes

- 1. The parameter *fetch* cannot be less than 300m.
- 2. The parameter *distrib* need not be set for the first HRU, i.e. the lowest vegetation height. When the snow is redistributed, the *distrib* values for the HRU's not filled to their vegetation height are summed and each HRU receives its share. The sum of distributions need not add to 1.0.

E.g. if the transport D is redistributed over HRUs: A, B and C with their *distrib* parameter having values of a, b and c respectively. The snow transport would be distributed as:

a*D/(a + b + c), b*D/(a + b + c) and c*D/(a + b + c). Note that any excess from a HRU filling to its vegetation height is deposited in the last HRU, i.e. the one having the tallest vegetation.

3. If the value of *distrib* is less than 0 for an HRU, all snow transport to its capacity is assumed to be caught by this HRU. The next HRU is the beginning of a new blowing snow regime.

4. This module can have a none zero value for transport and sublimation even if the vegetation height is greater than the snow cover depth.

5. Cumulative drift from the last HRU is not removed from the HRU and is redistributed within the HRU.

Typical values fo Zr.

landcover	ice	soil	open tundra	sparse shrub tundra	shrub tundra	dense shrub tundra	sparse forest
vegetation height (ht)	0.01	0.01	0.08	0.08	1.0	1.0	3.0
Zr	0.1	0.1	0.05	0.05	0.05	0.05	0.08

Code.

float Classsbsm::sublimation(void){

```
return 137.6*pow(hru_u[hh]/25.0, 5)/1000.0;
```

}

float Classsbsm::scale(void){

float

cond, // Thermal conductivity of air (W/m/K)

diff, // Diffusivity of water vapour in air (m2/s)

rsat, // Saturation density of water vapour (kg/m3)

tk; // Temperature (K)

float const ls = 2.838e6; // Latent heat of sublimation (J/kg)

float const m = 18.01; // Molecular weight of water (kg/kmole)

float const r = 8313.0; // Universal gas constant (J/kmole/K)

tk = hru_t[hh] + 273.0;

diff = 2.06e-5*pow(tk/273.0, 1.75);

```
rsat = m*611.15*exp(22.45*hru_t[hh]/tk)/(r*tk);
```

```
cond = 0.00063*tk + 0.0673;
```

```
return ((ls*m/(r*tk)) - 1.0)/(cond*(hru_t[hh]+273.0)) + 1.0/(ls*diff*rsat);
```

}

void Classsbsm::prob(void){

float

mean // Mean of cummulative normal distribution

```
,var // Standard deviation
```

```
,rho // Snow density (kg/m3)
```

```
,sd // Snow depth (m)
```

,us;

```
bool dry_snow = hru_t[hh] < 0.0;</pre>
```

```
mean = 0.365*hru_t[hh] + 0.00706*sqr(hru_t[hh]) +
0.91*log(snow_age[hh]) + 11.0;
var = 0.145*hru_t[hh] + 0.00196*sqr(hru_t[hh]) + 4.23;
if (!dry_snow) {
  mean = 21.0;
  var = 7.0;
}
rho = 240.;
if (SWE[hh] > 145.45)
  rho = 69.856*log(SWE[hh]) - 74.732;
sd = SWE[hh]/rho;
us = hru_u[hh];
if (sd < Ht[hh])
  us = us / sqrt(1. + 170.0*2*zr[hh]*(Ht[hh] - sd));
Prob[hh] = 1.0/(1.0 + exp(1.7*(mean - us)/var));
if (SWE[hh] <= wet_snow[hh]) {</pre>
  Prob[hh] = 1 / (1. + exp(1.7*(21.0 - us)/7.0));
```

if (us <= 7.0) Prob[hh] = 0.0;

}

if (sd <= 0.01) Prob[hh] = 0.0;

```
if ( dry_snow ){
    if (us <= 3.0)
    Prob[hh] = 0.0;
  }
  else{
    if (us <= 7.0)
    Prob[hh] = 0.0;
  }
}
void Classsbsm::run(void) {
float SumDrift, total, SWE_Max, trans;
  for (hh = 0; hh < nhru; hh++) {
    if(net_snow[hh] > 0.0) {
    SWE[hh] = SWE[hh] + net_snow[hh];
    cumSno[hh] = cumSno[hh] + net_snow[hh];
    snow_age[hh] = 1.0;
  }
  else
    snow_age[hh] += dt/3600;
  if(hru_t[hh] >= 0.0)
    wet_snow[hh] = SWE[hh];
```

else

```
wet_snow[hh] = min<float> (SWE[hh], wet_snow[hh]);
```

Drift[hh] = 0.0;

Subl[hh] = 0.0;

```
if(hru_u[hh] > 3.0 && SWE[hh] > 0.0) {
```

prob();

if(Prob[hh] > 0.0) {

Drift[hh] = Prob[hh]*transport()*dt/fetch[hh];

Subl[hh] = Prob[hh]*((1.0 - hru_rh[hh])/scale())* sublimation()*dt;

// handle insufficient snow pack

```
if(Drift[hh] + Subl[hh] > SWE[hh]) { Subl[hh] = SWE[hh] *
Subl[hh]/(Subl[hh] + Drift[hh]); Drift[hh] = SWE[hh] - Subl[hh]; } // end if
```

```
cumDrift[hh] += Drift[hh];
```

cumSubl[hh] += Subl[hh];

```
SWE[hh] = SWE[hh] - Subl[hh] - Drift[hh]; } }
```

} // for

```
// distribute drift
```

long LastN = 0;

for (int nn = 0; nn < nhru; ++nn) {

if(distrib[nn] >= 0.0 && nn+1 < nhru) continue;

SumDrift = 0.0;

```
for (int hh = LastN; hh <= nn; hh++)</pre>
```

```
if(distrib[nn] != 0.0)
```

SumDrift += Drift[hh]*hru_basin[hh];

```
if(SumDrift > 0.0){
```

```
for (int hh = LastN + 1; hh <= nn; hh++) {
```

if(hh == nn){ // last HRU or last HRU of group

SWE[hh] += SumDrift/hru_basin[hh];

```
cumDriftIn[hh] += SumDrift/hru_basin[hh]; }
```

else {

```
SWE_Max = SWEfromDepth(Ht[hh]);
```

```
if(SWE_Max > SWE[hh] && distrib[hh] > 0.0) {
```

```
total = 0.0;
```

```
for (int jj = hh; jj <= nn; jj++)
```

total = total + fabs(distrib[jj]);

trans = SumDrift*fabs(distrib[hh])/total/hru_basin[hh];

```
if(SWE_Max > SWE[hh] + trans)
```

```
SWE[hh] += trans;
```

else {

trans = SWE_Max - SWE[hh];

SWE[hh] = SWE_Max;

}

```
SumDrift -= trans*hru_basin[hh]; cumDriftIn[hh] += trans;
```

```
} // end if } // end if
```

```
\ // end for (hh) LastN = nn; \ // end if
```

```
} // end for (nn)
```

```
for (int hh = 0; hh < nhru; hh++) {
```

```
if(SWE[hh] > 0.0) const_cast<long*> (
```

```
inhibit_evap)[hh] = 1;
```

else

```
const_cast<long*> (inhibit_evap)[hh] = 0;
```

} // for

}

```
walmsley_wind (Walmsley et al., 1989)
```

This module is defined in Classwalmsley_wind and is parametric version of windflow model of Mason and Sykes (1979), referred as MS. The MS windflow model is a computational routine for estimating windflow over threedimensional topography and is based on Fourier transform techniques and has a division of the inner and outer flow regions. The MS windflow model is linearized and thus only applies to low hills; the model assumes neutral thermal stratification and uniform surface roughness within the simulation region. The MS model is potentially computational costly and Walmsley et al. (1989) derived a simple parametric version of the MS model for estimating wind speed variation induced by small-scale topographic features. These topographical features include 2D hills (ridges), 3D hills, 2D escarpments, 2D rolling terrain, 3D rolling terrain, and flat terrain. Users can define their simulation topographic features from coefficients 'A' and 'B'.

Observations

• none.

Variables

- hru_Uadjust (m/s) adjusted wind speed due to topography.
- hru_Uchange (m/s) amount of change in wind speed due to topography.
- WR () wind ratio between adjusted wind speed and reference wind speed.

Parameters

- Zwind (m) wind instrument height.
- A () coefficient for wind speed change due to topography. Values: 0.0 flat terrain, 2.5 2D escarpments, 3.0 2D hills, 3.5 2D rolling terrain, 4.0 3D hills, 4.4 3D rolling terrain.
- B () coefficient for wind speed change due to topography. Values: 0.0 flat terrain, 0.8 2D escarpments, 1.1 3D rolling terrain, 1.55 2D rolling terrain, 1.6 3D hills, 2.0 2D hills.

- L (m) upwind half-width at half height. It is the length from the centre of a hill with height, h to the slope with half height of the hill, h/2. It is also the length from the centre of a depression with depth, h to the slope with half depth of the depression, h/2.
- obs_elev (m) measurement altitude.
- hru_elev (m) altitude.

Variable Inputs

• hru_u (m/s) (obs)

Notes

Parameter *L* is upwind half-width at half height. It is the length from the centre of a hill with height, *h* to the slope with half height of the hill, h/2. It is also the length from the centre of a depression with depth, *h* to the slope with half depth of the depression, h/2.

wpe2.jpg (6658 bytes)

Reference

Mason, P.J. and Sykes, R.I. 1979. Flow over an isolated hill of moderate slope.

Quarterly Journal of the Royal Meteorological Society

105: 383-395.

Walmsley, J.L., Taylor, P.A. and Salmon, J.R. 1989. Simple guidelines for estimating windspeed variations due to small-scale topographic features – an update. *Climatological Bulletin* **23**: 3-14.

intcp

This module will handle summer and winter interception when the algorithms are finalized. At present it is included to handle the change of names required across the canopy because of canopy storage and evaporation. It is in effect a *null* module.

Observations

• none

<u>Variables</u>

- net_rain (mm/int) the amount of rainfall received by the ground surface under the canopy in time step, int. It is equal to the amount of rain received at the top of the canopy less the amount intercepted by the vegetation.
- cumnet_rain (mm) cummulatve rainfall.
- net_snow (mm/int) the amount of snowfall received by the ground surface under the canopy in time step, int. It is equal to the amount of snowfall received at the top of the canopy less the amount intercepted by the vegetation.
- net_p (mm/int) the total amount of precipitation (rain + snow) received by the ground surface under the canopy in time step, int. It is equal to the amount of precipitation the top of the canopy less the amount intercepted by the vegetation.

Parameters

- basin_area (km²) basin area.
- hru area (km²) HRU area.

Variable Inputs

- hru_rain (mm/int) HRU rain from module obs.
- hru_snow (mm/int) HRU snow from module obs.
- hru_p (mm/int) HRU precipitation from module obs.

Similar Modules.

- brushintcp.
- rutter.

Notes

CanopyClearing.

This module defined in ClassCanopyClearing, models net all-wave radiation at the snow surface under a needleleaf forest canopy from the observed incoming short-wave radiation and the Satterlund (1979) modified form of the Brunt equation. The canopy interception is modelled in summer using 'sparse' Rutter interception model (i.e. Valente 1997) and in winter a coupled forest snow interception and sublimation routine after Hedstom & Pomeroy / Parviainen & Pomeroy/Ellis:

Observations

- Qsi (W/m2) incident short-wave. *basic module and variation 1 i.e. CanopyClearing or CanopyClearing#1
- Qli (W/m2) incident long-wave. *basic module and variation 2 i.e. CanopyClearing or CanopyClearing#2

Observation Variable.

- Ts (°C) snow surface temperature.
- Qnsn (W/m2) net all-wave at snow surface.
- Qsisn (W/m2) incident short-wave at snow surface.
- Qlisn (W/m2) incident long-wave at snow surface.
- Qlosn (W/m2) -reflected long-wave at surface.

Variables

- drip_cpy (mm/int) canopy drip.
- direct_rain (mm/int) direct rain through canopy.
- net_rain (mm/int) direct rain + drip.
- cum_net_rain (mm) cumulative net_rain.
- Subl_Cpy (mm/int) canopy snow sublimation.
- cum_Subl_Cpy (mm) cumulative canopy snow sublimation.
- rain_load (mm) canopy rain load.
- snow_load (mm) canopy snow load.
- direct_snow (mm/int) direct snow through canopy.
- SUnload (mm/int) canopy snow unloaded.

- net_snow (mm/int) hru_snow less snow interception.
- Cum_net_snow (mm) cumulative net_snow.
- u_FHt (m/s) wind speed at canopy top.
- cc () canopy coverage.
- intcp_evap (mm/int) evaporation from the canopy.
- cum_intcp_evap (mm) cumulative canopy evaporation.
- Pevap (mm/int) Priestley/Taylor canopy evaporation when snowcover developed.
- k () extinction coefficient.
- Tauc () short-wave transmissivity through the canopy.
- ra (s/m) resistance.
- Pa (kPa) atmospheric pressure calculated from the elevation.
- Qnsn_var (W/m2) net all-wave at snow surface.
- Qsisn_var (W/m2) incident short-wave at snow surface.
- Qlisn_var (W/m2) reflected long-wave at snow surface.

Parameters

- CanopyClearing () canopy/clearing 0/1.
- Ht (m) canopy height.
- AlbedoCanopy () initial average HRU snow cover albedo. Only use by canopy. However should be set to 0.0 for clearing. See note below.
- hru_elev (m) HRU mean elevation.
- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- Z0snow (m) roughness length.
- Zref (m) temperature measurement height.
- Zwind (m) wind measurement height.
- LAI (m2/m2) leaf area index.
- inhibit_evap (flag) 0/1 enable/inhibit.
- Sbar (kg/m²) maximum canopy snow interception load.
- Zvent (m) ventilation wind speed height.
- unload_t (°C) ice-bulb temperature when canopy snow is unloaded as snow. Only use by canopy.
- unload_t _water(°C) ice-bulb temperature when canopy snow is unloaded as water. Only use by canopy.

Variable Inputs
- hru_t (°C) (*) air temperature.
- hru_rh (%) (*) relative humidity.
- hru_ea (kPa) (*) vapour pressure. Often calculated by an observation filter from t and rh.
- hru_u (m/s) (*) average wind speed over time step.
- SWE (mm) (*) snow water equivalent.
- SolAng (radians) (*) from global or similar module.
- hru_rain (mm/int) HRU rain from module obs.
- hru_snow (mm/int) HRU snow from module obs.
- hru_evap (*) (mm/int). Maximum evaporation when there is no snowcover.
- QliVt_var (W/m2) reflected long-wave at snow surface. *variation 1 and 3 i.e. CanopyClearingGap#1 or #3
- QsiS_var (W/m2) incident short-wave at snow surface. *variation 2 and 3 i.e. CanopyClearingGap#2 or #3

Notes.

The parameter Ht is used for the canopy height (vegetation) which is consistent with CRHM..

A parameter AlbedoCanopy is used for the canopy albedo. This may be handled using an external module or macro if the value changes with the season..

There is a problem.with the application of AlbedoCanopy. It modifies Qnsn for canopy and clearing as follows:

Qnsn[hh] = Qlisn[hh] - Qlosn[hh] + Qsisn[hh]*(1.0 - AlbedoCanopy[hh]) // for both canopy and clearing

AlbedoCanopy should be set to 0.0 for clearing. Since AlbedoCanopy for canopy is low (say 0.1) the error is usually very small if this is inadvertently used for clearing..

The canopy Tauc is not corrected to handle the change of radiation extinction path length for slope and aspect effects. Refer to the module CanopyClearingGap.

CanopyClearingGap.

This module defined in ClassCanopyClearingGap, models net all-wave radiation at the snow surface under a needleleaf forest canopy from the observed incoming short-wave radiation and the Satterlund (1979) modified form of the Brunt equation. The canopy interception is modelled in summer using 'sparse' Rutter interception model (i.e. Valente 1997) and in winter a coupled forest snow interception and sublimation routine after Hedstom & Pomeroy / Parviainen & Pomeroy/Ellis:

Observations

- Qsi (W/m2) incident short-wave. *basic module and variation 1 i.e. CanopyClearingGap or CanopyClearingGap#1
- Qli (W/m2) incident long-wave. *basic module and variation 2 i.e. CanopyClearingGap or CanopyClearingGap#2

Observation Variable.

- Ts (°C) snow surface temperature.
- Qnsn (W/m2) net all-wave at snow surface.
- Qsisn (W/m2) incident short-wave at snow surface.
- Qlisn (W/m2) incident long-wave at snow surface.
- Qlosn (W/m2) reflected long-wave at snow surface.

Variables

- drip_cpy (mm/int) canopy drip.
- direct_rain (mm/int) direct rain through canopy.
- net_rain (mm/int) direct rain + drip.
- cum_net_rain (mm) cumulative net_rain.
- Subl_Cpy (mm/int) canopy snow sublimation.
- cum_Subl_Cpy (mm) cumulative canopy snow sublimation.
- rain_load (mm) canopy rain load.
- snow_load (mm) canopy snow load.
- direct_snow (mm/int) direct snow through canopy.
- SUnload (mm/int) canopy snow unloaded.

- net_snow (mm/int) hru_snow less snow interception.
- Cum_net_snow (mm) cumulative net_snow.
- u_FHt (m/s) wind speed at canopy top.
- cc () canopy coverage.
- intcp_evap (mm/int) evaporation from the canopy.
- cum_intcp_evap (mm) cumulative canopy evaporation.
- Pevap (mm/int) Priestley/Taylor canopy evaporation when snowcover developed.
- k () extinction coefficient.
- Tauc () short-wave transmissivity through the canopy.
- ra (s/m) resistance.
- Pa (kPa) atmospheric pressure calculated from the elevation.
- Qnsn_var (W/m2) net all-wave at snow surface.
- Qsisn_var (W/m2) incident short-wave at snow surface.
- Qlisn_var (W/m2) reflected long-wave at snow surface.

Parameters

- CanopyClearing () canopy/clearing/gap 0/1/2.
- Ht (m) canopy height.
- AlbedoCanopy () initial average HRU snow cover albedo. Only use by canopy.
- hru_elev (m) HRU mean elevation.
- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- Z0snow (m) roughness length.
- Zref (m) temperature measurement height.
- Zwind (m) wind measurement height.
- LAI (m2/m2) leaf area index.
- inhibit_evap (flag) 0/1 enable/inhibit.
- Sbar (kg/m²) maximum canopy snow interception load.
- Zvent (m) ventilation wind speed height.
- unload_t (°C) ice-bulb temperature when canopy snow is unloaded as snow. Only use by canopy selection.
- unload_t _water(°C) ice-bulb temperature when canopy snow is unloaded as water. Only use by canopy selection.
- Gap_diameter (m) diameter of gap. Only use by gap selection.
- Surrounding_Ht (m) height of canopy around gap. Only use by gap selection.

Variable Inputs

- hru_t (°C) (*) air temperature.
- hru_rh (%) (*) relative humidity.
- hru_ea (kPa) (*) vapour pressure. Often calculated by an observation filter from t and rh.
- hru_u (m/s) (*) average wind speed over time step.
- SWE (mm) (*) snow water equivalent.
- SolAng (radians) (*) from global or similar module.
- cosxs (radians) (*) from global or similar module.
- cosxsflat (radians) (*) from global or similar module.
- hru_rain (mm/int) HRU rain from module obs.
- hru_snow (mm/int) HRU snow from module obs.
- hru_evap (*) (mm/int). Maximum evaporation when there is no snowcover.
- QliVt_var (W/m2) reflected long-wave at snow surface. *variation 1 and 3 i.e. CanopyClearingGap#1 or #3
- QsiS_var (W/m2) incident short-wave at snow surface. *variation 2 and 3 i.e. CanopyClearingGap#2 or #3

Notes.

The parameter Ht is used for the canopy height (vegetation) which is consistent with CRHM..

A parameter AlbedoCanopy is used for the canopy albedo. This may be handled using an external module or macro if the value changes with the season..

The canopy Tauc has been corrected to handle the change of radiation extinction path length for slope and aspect effects.

Originally:

```
Tauc = exp(-k*LAI_)
```

Now:

Tauc = exp(-k*LAI_*1/cosxs)

albedo (Gray and Landine, 1987)

This module is defined in Classalbedo and estimates the snow albedo throughout the winter and into the melt period. It also estimates the beginning of melt which initiates the snowmelt in the module ebsm. The algorithm developed uses daily values of net radiation, maximum and minimum daily temperatures, snowfall and snow-cover SWE.

Observations

• none

Variables

- albedo () snowcover albedo. State variable.
- meltflag () indicates start of melt. 0/1 for false/true. State variable.
- winter () indicates summer and winter events. 0/1. State variable.
- net_snowD (mm/d) daily snow fall.
- newsnowcnt (d) snow age count. Name is confusing as the value is the number of days since the last snow fall.

Parameters

- Albedo_bare () albedo for bare soil. Advance parameter.
- Albedo_snow () albedo for fresh snow. Advance parameter.
- hru_lat (°) latitude. Adjusts empirical Julian constants for Northern/Southern Hemisphere.

Variable Inputs

- hru_tmax (obs)
- hru_tmin (obs)
- hru_newsnow (obs)()
- QdroD (global) (MJ/m^2*d)
- SWE (pbsm)(mm)
- net_snow (intcp)(mm/int)

Notes

winter

The logical flag *winter* differentiates the major spring melt from fall and late spring snowfalls and associated melt events. *Winter* is set true between Julian days 300 and 2 when the SWE of the last HRU is greater than 5 mm. *Winter* is also set true when the daily maximum temperature is less than -6° C and the diagnostic variable Qnc is less than 1.0. *Winter* is set false when any HRU has no snowcover, i.e. SWE = 0.0.

meltflag

The logical flag *meltflag* indicates the days when melt is possible. It is always true when there is snowcover and *winter* is false. Otherwise it is only set when one of the melt criteria is met,

- 1. the daily minimum is greater than -4.0°C
- 2. Qnc > 1.0 AND the daily maximum > 0.0° C
- 3. Qnc > -0.5 AND the daily maximum > MT

where Qnc = -0.371 + 5.22*QdroD[hh]*(1 - A[hh]) and is a measure of incoming short-wave radiation,

and MT = -0.064*jday + 6.69 and is a probability index which increases as the radiative components increase as the year progresses.

newsnowcount

This counter is set to 0 after every snowfall and is incremented every day afterwards until amother snowfall occurs.

Albedo relationships.

At the onset of winter the albedo is set to 0.8. After a snowfall of greater than 5 cm depth (equivalent to a change in SWE of 0.25 mm, assuming a snow density of 0.005 kg/m3)

albedo_param.

This module is defined in Classalbedoparam. It sets the variable Albedo to the value of the parameter Albedo_value.

Observations

• none

Variables

• Albedo () - Tthis variable is set to value of Albedo_value.

Parameters

• Albedo_value () - parameter value.

Variable Inputs

• none.

Notes.

The module *albedo_obs* is more flexible as it does the same as this module but can also read albedo observations.

albedo_obs.

This module is defined in Classalbedoobs. It read a series of albedo observations into the variable Albedo.

Observations

• Albedo_obs () - average albedo over the time interval. Optional.

Variables

• Albedo () - Observation values.

Parameters

• Albedo_value () - default value. Used if observation are not available and initially until first available observation.

Variable Inputs

• none.

Notes

If the Observation is not available Albedo takes the value of Albedo_value.

global

Defined in ClassGlobal. This module calculates the theoretical interval shortwave direct and diffuse solar radiation. The routine handles GSL and ASL. There is an adjustment for elevation and transmissivity applicable to the region adjacent to the triangle, Edmonton, Winnipeg and Bad Lake. The interval and daily values are calculated by accumulating the 10 minute calculated values. The maximum number of daily sunshine hours is calculated. The theoretical directbeam component of solar radiation is developed from an expression proposed by Garnier and Ohmura (1970). The solar angle is output for use in other modules. Extra variables have been added to help in determining if the module is malfunctioning

Observations

• none

Variables

- Qdro (W/m^2) clear-sky direct.
- Qdfo (W/m^2) clear-sky diffuse.
- Qdflat (W/m^2) clear-sky direct + diffuse on a horizontal surface. Always independent of slope and aspect.
- QdflatE (W/m^2) extraterrestrial radiation, i.e. no atmosheric attenuation. Always independent of slope and aspect.
- QdroD (MJ/m^2) daily clear-sky direct.
- QdfoD (MJ/m^2) daily clear-sky diffuse.
- QdflatD (MJ/m²) daily clear-sky direct on a horizontal surface. Now includes the diffuse.
- QdroDext (W/m^2) daily extraterrestial direct. Always independent of slope and aspect.
- SolAng (r) solar angle in radians.
- SunMax (h) maximum sunshine hours.
- pQdro (MJ/m^2*int) clear-sky direct.
- pQdfo (MJ/m²*int) clear-sky diffuse.
- pSol(r) solar angle in radians.

Parameters

- hru_lat (°) latitude. Positive value indicates northern hemisphere, negative value southern hemisphere.
- hru_elev (m) altitude.
- hru_GSL (°) ground slope increasing the slope positively, tilts the plane to the north with ASL = 0.
- hru_ASL (°) azimuth. 0/90/180/270 north/east/south/west facing for positive GSL.
- Time_Offset (h) solar time offset from local time.

Variable Inputs

• none

Notes

Calculation of Diffuse.

A simple means of calculating the dffuse clear-sky radiation is given by List (1968) as:

diffuse = 0.5((1-aw-ac)Qa - Id) where

aw = radiation absorbed by water vapour (7%)

ac = radiation absorbed by ozone (2%)

Qa = extra-terrestrial radiation on a horizontal surface at the outer limit of the earth's atmosphere.

Id = direct clear-sky radiation reaching the earth's surface on a horizontal surface.

Optical airmass.

The relative path length of the direct solar beam radiance through the atmosphere. When the sun is directly above a sea-level location the path length is defined as airmass 1 (AM 1.0). AM 1.0 is not synonymous with solar noon

because the sun is usually not directly overhead at solar noon in most seasons and locations. When the angle of the sun from zenith (directly overhead) increases, the airmass increases approximately by the secant of the zenith angle. A better calculation (*Kasten, F. and A. T. Young (1989). Revised optical air mass tables and approximation formula.* **Applied Optics 28** (22), 4735-4738) follows:

 $m = 1.0 / [\cos(Z) + 0.50572 * (96.07995 - Z)^{-1.6364}]$

where Z is the solar zenith angle.

Annandale

Defined in ClassAnnandale. The purpose of this module is to estimate incoming short wave radiation from daily minimum and maximum temperatures when no other radiation observations are available. Annandale et al. (2001) developed a simple emipirical method for estimating transmittance from the daily range of air temperature.

Observations

• none

Variables

- TauA () transmittance calculated from the Annandale method.
- hru_SunAct (h) calculated sunshine hours.

Parameters

- krs () location index, interior = 0.16 and coastal = 0.19.
- hru_elev (m) HRU elevation.

Variable Inputs

- hru_tmax (*) (°C) daily maximum temperature.
- hru_tmin (*) (°C) daily minimum temperature.
- QdroDext (*) (MJ/m^2*d) daily ExtraTerrestrial direct.
- QdflatD (*) (MJ/m^2*d) daily clear-sky Qdro (with diffuse) on horizontal surface.
- QdflatE (*) (W/m²) 'Qdro' on horizontal surface, no atmosheric attenuation.
- Qdro (*) (W/m²) clear-sky direct with GSL and ASL applied.
- Qdfo (*) (W/m^2) clear-sky diffuse with GSL and ASL applied.
- QdfoD (*) (MJ/m^2*d) daily average clear-sky diffuse.
- SunMax (*) (h) maximum sunshine hours.

Observation Variables

- QsiA (W/m^2) interval synthesized short-wave derived from Annandale method.
- QsiD (W/m^2) daily synthesized short-wave derived from Annandale method.
- QsiS (W/m^2) synthesized short-wave on slope derived from global Qdro and Qdfo, and the Annandale method.

netall

This module defined in Classnetall, models net all-wave radiation from the calculated short-wave radiation and the Brunt equation. Climatic data required are temperature, vapour pressure and actual sunshine hours.

Classnetall is an enhancement of the original class which provides interval radiation for the the day to allow modules to partition calculated daily evaporation over the day to intervals with positive net radiation. These calculations must be done at the beginning of the day to be available throughout the day.

Observations

none

Variables

- net $(MJ/m^2 \Delta t)$ average all-wave net over time step.
- Rn (mm/m² Δt) average all-wave net over time step.
- netD (MJ/m² day) average all-wave net over time day.
- RnD (mm/m² day) average all-wave net over time day.
- RnD_Pos (mm/m² day) sum of positive all-wave net over time day.
- cum_net (MJ/m²) cummulative all-wave net.

Parameters

• Albedo () - areal albedo. // pseudo

Variable Inputs

- hru_t (obs)
- hru_ea (obs)
- Tday (obs) t interval values for entire day.
- EAday (obs) ea interval values for entire day.
- hru_SunAct (obs)

- SunMax (global)
- Qdro (global)
- Qdfo (global)
- pQdro (global)
- pQdfo (global)
- albedo (albedo)

Returns

• returns (netlongwave + netshortwave*(1-hru_alb))/Freq

where:

```
netlongwave = -0.85 + 0.97*SB*pow(hru_t+273.0, 4)*
(-0.39+0.093*sqrt(hru_ea)*(0.26+0.81*(hru_SunAct/SunMax))
```

if(hru_SunAct)

```
netshortwave = (0.024+0.974*pow(hru_SunAct/SunMax,
1.35))*Qdro*0.0864 +(2.68+2.2*
(hru_SunAct/SunMax)-3.85*sqr(hru_SunAct/SunMax))*Qdfo*0.0864
```

else

```
netshortwave = (0.024*Qdro + 2.68*Qdfo)*0.0864
```

Freq is the number of intervals in the day.

long

This module defined in Classlong calculates long-wave radiation. Incoming Longwave radiation to Melting Snow: Observations, Sensitivity and Estimation in Northern Environments (Sicart et al., 2005?).

Observations

• Qsi (W/m^2) - incident short-wave.

Observation Functions

• QsiDtot(W/m^2*d) - daily total (cumulative) of average interval incident short-wave fluxes, Qsi.

observationVariables

• Qli# (W/m^2) - calculated incoming long-wave.

Variables

- tau () atmospheric transmissivity.
- Long (W/m^2) calculated incoming long-wave. Same as generated observation.

Parameters

• none.

Variable Inputs

- hru_t (°C) air temperature.
- hru_ea (kPa) vapour pressure.
- hru_rh (%) relative humidity.
- QdroDext (global)(MJ/m^2*d) daily extraTerrestial direct.
- QdroD (global) (MJ/m^2*d) daily clear-sky direct.

Notes

tau = QsiDtot*(864000/10E6)/QdroDext.

Qli = $1.24*(e/T)^{(1/7)}(1 + 0.44*RH - 0.18*tau)sbcT^4$ where RH lie in the range of 0 and 1 and T is °K.

The declared observation, Qsi# cannot be used as input as the daily total cannot be calculated. If an attempt is made by the module to use a declared observation it will be automatically converted to the simple observation, i.e. Qsi# converted to Qsi.

History.

March 2008 - temperature, vapour pressure and relative humidity changed from using observations to the HRU variables.

calcsun

This module is defined in Classcalcsun and can estimate sunshine hours from the the incoming short-wave observation Qsi and the values QdroD, QdfoD and SunMax calculated by the module global using the HRU's latitude.

Observations

- Qsi (W/m²) average incident short-wave flux for time interval (optional). Used only if sunshine hours are not available.
- SunAct (h) sunshine hours (optional). When not available the sunshine hours are estimated from Qsi, if available.

Observation Functions

• Qsitot(W/m^2*d) - daily total (cumulative) of average interval incident short-wave fluxes, Qsi.

Variables

• hru_SunAct (h) - calculated sunshine hours.

Parameters

• none

Variable Inputs

- QdfoD (global) (MJ/m^2*d)
- QdflatD (global) (MJ/m^2*d)
- SunMax (global) (h)

Notes.

When the actual sunshine hours are not available as an observation, an estimate is made from the following,

hru_SunAct = (daily sum of Qsi*WtoMJ_D/Freq - QdfoD) * (theoretical maximum sunshine hours)/(QdflatD - QdfoD) where WtoMJ_D = 86400/10^6 and Freq is intervals/day,

otherwise the observation SunAct is used.

NeedleLeaf.

This module defined in Classneedleleaf, models net all-wave radiation at the snow surface under a needleleaf forest canopy from the observed incoming short-wave radiation and the Satterlund (1979) modified form of the Brunt equation. Other climatic data required are air temperature, relative humidity/vapour pressure and windspeed.

Observations

- Qsi (W/m2) incident short-wave.
- Qli (W/m2) incident long-wave. Optional.

Observation Variable.

- Ts (°C) snow surface temperature.
- Qnsn (W/m2) net all-wave at snow surface.
- Qsisn (W/m2) incident short-wave at snow surface.
- Qlisn (W/m2) incident long-wave at snow surface.
- Qlisn (W/m2) -refected long-wave at snow surface.

Variables

- k () extinction coefficient.
- Tauc () short-wave transmissivity through the canopy.
- ra (s/m) resistance.
- Pa (kPa) atmospheric pressure calculated from the elevation.

Parameters

- Ht (m) canopy height.
- Albedo () initial average HRU snow cover albedo.
- hru_elev (m) HRU mean elevation.
- Z0snow (m) roughness length.
- Zref (m) temperature measurement height.
- Zwind (m) wind measurement height.
- LAI (m2/m2) leaf area index.

Variable Inputs

- hru_t (°C) (*) air temperature.
- hru_rh (%) (*) relative humidity.
- hru_ea (kPa) (*) vapour pressure. Often calculated by an observation filter from t and rh.
- hru_u (m/s) (*) average wind speed over time step.
- SWE (mm) (*) snow water equivalent.
- albedo (albedo) (*)
- SolAng (radians) (*) from global or similar module.

Notes.

```
Pa[hh] = 101.3*pow((293.0-0.0065*hru_elev[hh])/293.0, 5.26); // kPa
```

```
Exposure = Ht[hh] - DepthofSnow(SWE[hh]); /* depths(m) SWE(mm)
```

```
if(Exposure < 0.0) Exposure = 0.0
```

```
LAI_ = LAI[hh]*Exposure/Ht[hh]
```

```
Vf_ = Vf[hh] + (1.0 - Vf[hh])*sin((Ht[hh] - Exposure)/Ht[hh]*M_PI_2)
```

```
if(beta[hh] > 0.001) {
```

```
k[hh] = 1.081*beta[hh]*cos(beta[hh])/sin(beta[hh]);
```

```
Tauc[hh] = exp(-k[hh]*LAI_[hh]);
```

}

else {

```
k[hh] = 0.0; Tauc[hh] = 0.0;
```

```
}
rho[hh] = Pa[hh]*1000/(Rgas*(t[0] + 273.15));
```

float U1 = max(u[0], 1.0e-3); // Wind speed (m/s)

ra[hh] = sqr(log(z1[hh]/z0[hh]))/(sqr(kappa)*U1);

delta[hh] = 0.622*Ls*Qstar[hh]/(Rgas*sqr(t[0] + 273.15));

float q = (rh[0]/100)*Qs(Pa[hh], T1); // specific humidity (kg/kg)

Ts[hh] = T1 + (emiss*(Qli[Qlimax] - sbc*pow(T1, 4)) + Ls*(q - Qs(Pa[hh], T1))*rho/ra[hh])/(4*emiss*sbc*pow(T1, 3) + (Cp + Ls*delta)*rho/ra[hh])

// Canopy temperature is approximated by the air temperature.

 $\label{eq:Qn[hh] = Vf_[hh]*Qli[Qlimax] + (1.0 - Vf[hh])*emiss_c*sbc*pow(T1, 4.0) - emiss*sbc*pow(Ts[hh] + Tm, 4.0) + Qsi[0]*Tauc[hh]*(1-Albedo[hh])$

Constants

sbc = 5.67E-8 (W/m2/k4) Stephan-Boltzmann constant.

Ls = 2.845e6 (J/kg) latent heat of sublimation.

Lf = 0.334e6 (J/kg) latent heat of fusion.

Rgas = 287.0 (J/kg/K) gas constant for dry air.

rho = 1.23 (kg/m3) density of dry air.

Cp = 1005 (J/kg/K) volumetric heat capacity of dry air.

Tm = 273.15 (K) melting point.

Kevin

This module is defined in ClassKevin. Snow accounting is handled in PBSM using snow fall, transport and sublimation.

Observations

• hru_Qn (obs) - net radiation (W/m^2). Optional.

<u>Variables</u>

- albedo () areal albedo.
- snowmelt (mm/int) depth/amount of snowmelt in time step, ∆t.
- snowmeltD (mm/d) daily snowmelt.
- cumsnowmelt (mm) index day melt.
- sca () snow covered fraction. State variable.
- winter () winter = 0/1 false/true. State variable.
- SnowStat () EARLY/MATURE/HOLD 0/1/2. State variable.

Parameters

- Asnow1 () early snow albedo,. fresh snow, newly-fallen snow, snow at the start of ablation or "active melt".
- Asnow2 () mature snow albedo, wet snow, metamorphosed snow.
- Asoil () soil albedo.
- cv () coefficient of variation.
- tfactor (mm/d*/°C) degree-day melt factor.
- nfactor (mm*m^2/MJ*d) net radiation factor. Converts daily net radiation flux (MJ/m2) to an equivalent average depth of water
- meltthresh(mm/d) amount of daily melt that must be exceeded in order for a snowcover to become a mature pack.
- hru_lat (°) latitude. Adjusts empirical Julian constants for Northern/Southern Hemisphere.

Variable Inputs

• hru_t (obs) - air temperature (°C) from module obs.

- hru_tmean (obs) air temperature (°C) from module obs.
- hru_eamean (obs) mean daily vapour pressure (kPa) from module obs.
- hru_SunAct(obs) actual sunshine hours (h).
- SunMax (global) maximum sunshine hours (h).
- Qdro (global) calculated clear-sky direct (W/m^2).
- Qdfo (global) calculated clear-sky diffuse (W/m^2).
- SWE (pbsm) Snow water equivalent (mm).

Internal Variables

- SWEpeak (mm) maximum SWE before melt.
- SWElast (mm) last SWE before new snow fall during melt.
- daymelt (mm) cumulated melt during a day. Used to determine when to go from the Early to the Mature phase.

Melt Algorithms

• tfactor OR nfactor greater than zero.

melt (mm/int) = (hru_t*tfactor + hru_Qn*nfactor*WtoMJ_D)/Freq
individual terms used when positive.

• tfactor AND nfactor both equal to zero.

```
netlong = -0.85 + 0.97*SB*pow(hru_tmean+273.0, 4)*
(-0.39+0.093*sqrt(hru_eamean))*(0.26+0.81*(hru_SunAct/SunMax))
```

```
if(hru_SunAct > 0.0)
```

```
shortw = (0.024+0.974*pow(hru_SunAct/SunMax,
1.35))*Qdro*WtoMJ_D +(2.68+2.2*
(hru_SunAct/SunMax)-3.85*sqr(hru_SunAct/SunMax))*Qdfo*WtoMJ_D
```

else

shortw = (0.024*Qdro + 2.68*Qdfo)*WtoMJ_D

net = (shortw*(1.0-albedo) + netlong)/Freq; // MJ/day to MJ/interval.

```
where albedo = asnow1 for EARLY and = asnow2 for MATURE
```

melt (mm/int) = net/Hf // Hf = 0.3336 Latent heat of fusion, MJ/kg.

Snow Covered Area (SCA)

This is calculated by a routine <u>SWE_prob</u> from Shook 1995. The lognormal probability density function is used to describe the frequency distribution of the snow water equivalent. The routine calculates the exceedence probability for a value of SWE (Melt), i.e., it is the probability of a given SWE-value being exceeded. The value SWEpeak is the arithmetic mean of the snow pack before the onset of melt.

Model Flow Structure

The season is handled as three separate states

- 1. EARLY snow accumulation during early winter.
- 2. MATURE snow accumulation in spring after a significant melt.
- 3. HOLD occurs during the Mature state when a heavy snow fall occurs causing the snow pack to increase again.

EARLY: is the default state entered in the fall. SCA is 100% and the snow albedo is set to "Asnow1". After the spring melt it is re-entered when SCA is less than 1%.

MATURE: is entered when a melt has exceeded the value of "meltthresh". The albedo is set to "Asnow2". The SCA is calculated from the peak SWE (SWEpeak) and the remaining snow water equivalent (SWEpeak - SWElast).

HOLD: is entered from the Mature state when the SWE has exceeded the last interval SWE (SWElast). It will stay in this state until the SWE becomes less than value when the state was entered (SWElast) or SWE becomes greater than SWEpeak. SCA is 100% and the snow albedo is set to "Asnow1"

Status

• incomplete

Slope_Qsi.

This module estimates Qsi for a slope from the measured incoming short-wave Qsi on the level. The ratio of measured Qsi and the calculated clear sky direct and diffuse on a horizontal plane is used to adjust the calculated clear sky value on the slope.

Observations

• Qsi (W/m^2) - incident short-wave measured on a level plane.

Variables

- Short (W/m^2) Calculated Qsi for the slope equal to Qdro + Qdfo.
- ShortC (W/m²) Qsi corrected for slope from the ratio of (measured Qsi)/Qdflat.

Parameters

• none.

Variable Inputs

- Qdro (global) (W/m^2).
- Qdfo (global) (W/m^2).
- Qdflat (global)(W/m^2).

Observation Variables

• Qsi# (W/m^2) - Qsi corrected for slope.

Truncation note.

In this implementation there is truncation at either sunrise or sunset since the alogorithm is only active when the incoming net short wave radiation on the level is positive. This means that any incoming radiation due to the slope at before dawn and after dusk is ignored unless the incoming is positive on the

level.

evap (Granger, 1989; Granger and Pomeroy, 1997)

This module defined in Classevap, calculates the evapotranspiration using interval values from after snowmelt to fall snow cover.

It has been enhanced to partition calculated daily Granger evaporation over the day to intervals with positive net radiation.

Other evaporation algorithm available is Priestley-Taylor.

Observations

• none

Variables

- hru_evap (mm/int) average depth of evaporation from an HRU in a time step, *∆*t.
- hru_cum_evap (mm) cumulative evaporation calculated as the sum of interval estimates of hru_evap.
- hru_evapD (mm/d) daily evaporation calculated as the daily sum of interval estimates of hru_evap. Should equal evap_GrangerD
- evap_GrangerD (mm/d) *daily* evaporation calculated from daily means of t, u and vapour pressure.
- evap_G () relative evaporation from Granger.
- evap_D () relative drying power from Granger.
- hru_actet (mm/int) actual evapotranspiration over the HRU. Evaporation and evapotranspiration are limited by the amount of soil moisture available.
- hru_cum_actet (mm) cumulative actual evapotranspiration over the HRU.

Parameters

- evap_type (flag) Evaporation method for HRU, 0 Granger, 1 Priestley Taylor.
- Ht (m) crop height.
- hru_elev (m) altitude.
- basin_area (km^2) basin area.

- hru area (km^2) HRU area.
- F_Qg () fraction to ground flux. Qg = F_Qg*Rn.
- inhibit_evap (flag) 0/1 enable/inhibit.

Variable Inputs

The source of the variable is given in brackets after the name. A full description of the variable may be found there.

- hru_t (obs) (°C)
- hru_tmean (obs) (°C)
- hru_umean (*) (m/s)
- hru_eamean (obs) (kPa)
- Rn (*) (mm/m^2*int)
- RnD (*) (mm/m^2*d)
- RnD_Pos (*) (mm/m^2*d)

Returns

Daily calculation of Granger.

- Ea = $\underline{fdaily}(u, Ht)^*(e^*-ea)$
- D = Ea/(Ea+(Rn-Qg-Qs))
- $G = 1/(0.793 + 0.2 \exp(4.902 D) + 0.006 D$
- E = (delta(t)*G*(Rn-Qg-Qs) + gamma(Pa, t)*G*Ea)/(delta(t)*G + gamma(Pa, t))

where t, u and ea are the mean of interval values and e* is calculated using the mean daily temperature t.

Priestley Taylor

• 1.26*delta(t)*(Rn-Qg-Qs)/((delta(t) + gamma(Pa, t)) for positive incoming net radiation,

where values are interval. Evaporation is zero when incoming net radiation is ≤ 0.0 .

Atmospheric Pressure

• P = P0*exp((293.0-0.0065*elevation)/293.0, 5.26),

where P0 = 101.3.

Granger Daily Evaporation Partitioned over daily intervals when Rn is positive.

hru_evap = hru_evapD*Rn[t_n] / RnD_POS,

where RnD_POS is the sum of daily intervals when $Rn[\Delta t]$ is positive.

evapX - Evaporation calculated using Granger and Pomeroy, 1997, Priestley-Taylor and Penman-Monteith methods.

This module defined in ClassevapX, calculates the evapotranspiration from after snowmelt to fall snow cover.

It calculates the daily evapotranspiration using Granger and partitions it over the day to interval values using intervals with positive net radiation.

The interval values are calculated for the Priestley-Taylor and Penman-Monteith methods.

Observations

- RnObs (W/m^2) all-wave net radiation used by Penmon_Monteith and Dalton Bulk transfer method (optional).
- Qsi (W/m²) incident short-wave (optional).

Variables

- hru_evap (mm/int) average depth of evaporation from an HRU in a time step, *∆*t.
- hru_cum_evap (mm) cumulative evaporation calculated as the sum of interval estimates of hru_evap.
- hru_evapD (mm/d) daily evaporation calculated as the daily sum of interval estimates of hru_evap. Should equal evap_Granger_D etc.
- hru_actet (mm/int) actual evapotranspiration over the HRU. Evaporation and evapotranspiration are limited by the amount of soil moisture available.
- hru_cum_actet (mm) cumulative actual evapotranspiration over the HRU.
- Granger_D (mm/d) daily evaporation calculated from daily means of t, u and vapour pressure using Granger_D method.
- Priestley_Taylor_D (mm/d) daily evaporation calculated from daily means of t, u and vapour pressure using Priestley_Taylor method.
- rc (s/m) stomatal resistance (used by Penman-Monteith and Dalton).
- Pa (kPa) Atmospheric pressure.

Parameters

- evap_type (flag) Evaporation method for HRU, 0 = Granger, 1 = Priestley-Taylor, 2 = Penman-Monteith.
- Ht (m) crop height.
- hru_elev (m) altitude.
- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- F_Qg () fraction to ground flux. Qg = F_Qg*Rn.
- inhibit_evap (flag) 0/1 enable/inhibit.
- Zwind () wind measurement height (used by Penman-Monteith).
- rcs (s/m) stomatal resistance (used by Penman-Monteith).
- soil_type () HRU soil type (used by Penman-Monteith) [1->11]: sand/loamsand/sandloam/loam/siltloam/sasclloam/clayloam/siclloam/sandcl
- soil_Depth (m) depth of soil column (used by Penman-Monteith).
- Htmax (m) maximum vegetation height (used by Penman-Monteith).
- LAImax (m²/m²) maximum leaf area index (used by Penman-Monteith).
- LAImin (m²/m²) minimum leaf area index (used by Penman-Monteith).
- s () seasonal growth index (used by Penman-Monteith).
- PM_method () Penman-Monteith method (used by Penman-Monteith), 0 = RC min, 1 = LAI, 2 = bulk.

Variable Inputs

The source of the variable is given in brackets after the name. A full description of the variable may be found there.

- hru_t (obs) (°C)
- hru_tmean (obs) (°C)
- hru_u (*) (m/s)
- hru_umean (*) (m/s)
- hru_ea (obs) (kPa)
- hru_eamean (obs) (kPa)
- Rn (*) (mm/m^2*int)
- RnD (*) (mm/m^2*d)
- RnD_Pos (*) (mm/m^2*d)

Evaporation calculations Daily/Interval.

Interval observations or daily mean observations can be used in Priestley-

Taylor and Penman-Monteith methods. The selection is made by the parameters PT_Daily, PM_Daily and DB_Daily. When

Daily calculation of Granger.

- Ea = $\underline{fdaily}(u, Ht)^*(e^*-ea)$
- D = Ea/(Ea+(Rn-Qg-Qs))
- $G = 1/(0.793 + 0.2 \exp(4.902 D) + 0.006 D$
- E = (delta(t)*G*(Rn-Qg-Qs) + gamma(Pa, t)*G*Ea)/(delta(t)*G + gamma(Pa, t))

where t, u and ea are the mean of interval values and e* is calculated using the mean daily temperature t.

The Granger daily evaporation is partitioned over the daily intervals when Rn is positive.

hru_evap = hru_evapD*Rn[t_n] / RnD_POS,

where RnD_POS is the sum of daily intervals when $Rn[\Delta t]$ is positive.

Priestley Taylor

• 1.26*delta(t)*(Rn-Qg-Qs)/((delta(t) + gamma(Pa, t)) for positive incoming net radiation,

where values are interval. Evaporation is zero when incoming net radiation is ≤ 0.0 .

Atmospheric Pressure

• P = P0*exp((293.0-0.0065*elevation)/293.0, 5.26),

where P0 = 101.3.

evapD (Granger, 1989; Granger and Pomeroy, 1997)

This module defined in ClassevapD, calculates the evapotranspiration using daily values from after snowmelt to fall snow cover.

Other evaporation algorithm available is Priestley-Taylor.

Observations

• none

Variables

- hru_evapD (mm/d) average depth of evaporation from an HRU.
- hru_cum_evap (mm) cumulative evaporation calculated as the sum of the daily estimates of hru_evap.
- evap_G () relative evaporation from Granger.
- evap_D () relative drying power from Granger.
- hru_actet (mm/int) actual evapotranspiration over the HRU. Evaporation and evapotranspiration are limited by the amount of soil moisture available.
- hru_cum_actet (mm) cumulative actual evapotranspiration over the HRU.

Parameters

- evap_type (flag) Evaporation method for HRU, 0 Granger, 1 Priestley Taylor.
- Ht (m) crop height.
- hru_elev (m) altitude.
- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- F_Qg () fraction to ground flux. Qg = F_Qg*Rn.
- inhibit_evap (flag) 0/1 enable/inhibit.

Variable Inputs

- hru_tmean (obs) (°C)
- hru_umean (*) (°C)
- hru_eamean (obs) (°C)
- RnD (*) (mm/m^2*d)

Granger Daily calculation

- Ea = $\underline{fdaily}(u, Ht)*(e^{*}-ea)$
- D = Ea/(Ea+(Rn-Qg-Qs))
- $G = 1/(0.793 + 0.2 \exp(4.902 D) + 0.006 D)$
- E = (delta(t)*G*(Rn-Qg-Qs) + gamma(Pa, t)*G*Ea)/(delta(t)*G + gamma(Pa, t))

where t, u and ea are the mean of interval values and e* is calculated using the mean daily temperature t.

Priestley Taylor

• 1.26*delta(t)*(Rn-Qg-Qs)/((delta(t) + gamma(Pa, t))f or positive incoming net radiation,

where values are interval. Evaporation is zero when incoming net radiation is ≤ 0.0 .

Atmospheric Pressure

• P = P0*exp((293.0-0.0065*elevation)/293.0, 5.26),

where P0 = 101.3.

lake evaporation (Meyer Formula with revisions by PFRA)

This module defined in Classlake, calculates the monthly evaporation and distributes it uniformly over every interval of the month.

Observations

• none

Variables

- hru_evap (mm/int) average depth of evaporation from an HRU in a time step, *∆*t.
- hru_cum_evap (mm) cumulative evaporation calculated as the sum of interval estimates of hru_evap.
- lake_evap_month (mm) monthly evaporation from rh.
- hru_actet (mm/int) actual evapotranspiration over the HRU. Evaporation and evapotranspiration are limited by the amount of soil moisture available.
- hru_cum_actet (mm) cumulative actual evapotranspiration over the HRU.
- Va (mm) water vapour pressure, Vw*rh (Meyer).
- Vw (mm) air vapour pressure (Meyer).
- hru_t_mean (°C) avereage monthly temperature.
- hru_rh_mean () avereage monthly relative humidity.
- hru_u_mean (m/s) avereage monthlywind speed.
- hru_t_acc (°C*N) accumulated monthly temperature.
- hru_rh_acc (N) accumulated monthly relative humidity.
- hru_u_acc (m/s*N) accumulated monthly wind speed.

Parameters

- Meyer_C () Meyer coefficient.
- Meyer_B () Meyer monthly intercept coefficient.
- basin_area (km^2) basin area.
- hru area (km^2) HRU area.
- start_open_water () Julian date of start of lake evaporation.
- end_open_water () Julian date of end of lake evaporation.

Variable Inputs

The source of the variable is given in brackets after the name. A full description of the variable may be found there.

- hru_t (obs) (°C)
- hru_rh (obs) ()
- hru_u (obs) (m/s)

Equations.

hru_t_Mmean = hru_t_acc/N_intervals; // where N_intervals is the number of time steps in the month. hru_rh_Mmean = hru_rh_acc/N_intervals; hru_u_Mmean = hru_u_acc/N_intervals;

float Tw = 0.6*hru_t_Mmean + Meyer_B[ThisMonth-1];

```
Vw = pow(10,(-7.903*(373.16/(Tw+273.16)-1)+
(5.028*log10(373.16/(Tw+273.16)))
-(0.0000001382*(pow(10, (11.34*(1-(373.16/(Tw+273.16)))))-1))
+(0.008133*pow(10, (-3.491*((373.16/(Tw+273.16))-1)))))+3.006);
```

```
Va = Vw*hru_rh_Mmean/100.0;
```

lake_evap_month = 0.75002*Meyer_C*(Vw-Va)*(1+0.06214*hru_u_Mmean);

hru_evap = lake_evap_month/N_intervals; hru_cum_evap += hru_evap;

Note.

The months containing the Julian date of start of evaporation and end of operation must be complete months for the correct calculation of the monthly evaporation.

Ht_obs.

This module is defined in ClassHtobs. It reads a series of vegetation height observations into the parameter Ht. Parameters are normally constant during a model run. This module makes the parameter Ht follow the current value of the height in the observation file.

Observations

• Ht_obs (m) - average vegetation height over the time interval. Optional, when not available the parameter value Ht retains its initial.

Variables

• Ht_var (m) - Observation values as a variable for monitoring.

Parameters

• Ht (m) - has initial default value. The initial value is used if no observations are available or until the first available observation is read.

Variable Inputs

• none.

Notes.

The variable Ht_var is provided for debugging. Since parameters cannot be displayed by CRHM (assumed to be constant), Ht may be monitored by displaying the variable Ht_var.

ebsm (Gray and Landine, 1987)

This module is defined in Classebsm and is an energy-budget snowmelt model applicable to the Canadian Prairies.

Observations

• QnD (W/m^2 day) - net radiation. Optional.

Variables

- snowmeltD (mm/d) snowmelt.
- cumsnowmelt (mm) cummulative snowmelt.
- LW_ebsm (mm/d) liquid water in snowpack. State variable.
- u_ebsm (MJ/m^2*d) snowpack energy deficit. State variable.
- Qmelt (MJ/m^2*d) Qmelt = Qn+Qh+Qe+Qp.
- Qn_ebsm (MJ/m^2*d) net radiation.
- Qh_ebsm (MJ/m^2*d) sensible heat.
- Qe_ebsm (MJ/m^2*d) latent heat.
- Qp_ebsm (MJ/m^2*d) input from rainfall.
- Qe_Subl (mm/day) latent heat sublimation.
- cumQe_subl(mm) cummulative latent heat sublimation.

Parameters

- tfactor (mm/d*/°C) degree day melt factor.
- nfactor (mm*m^2/MJ*d) net radiation factor.
- delay_melt () inhibits any melt before this date. Only used when tfactor and nfactor are both zero. Default value not suitable for Southern Hemisphere
- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- Albedo () areal albedo.
- Qe_subl_SWE () 0 add to 'Qmelt' (original version), 1 take Qe_subl from SWE .

Variable Inputs

- hru_tmean (obs) (°C)
- hru_tmax (obs) (°C)
- hru_tmin (obs) (°C)
- hru_umean (obs) (m/s)
- hru_rhmean (obs) (%)
- hru_Sun_Act (obs) (hr)
- net_rain (*) (mm ∆t)
- QdroD (global) (MJ/m^2)
- QdfoD (global) (MJ/m^2)
- SunMax (global) (h)
- SWE (*) (mm)
- albedo (albedo) ()
- meltflag (albedo) ()

Useage.

tfactor and nfactor NOT equal to zero

For comparison there is an option to use the simplified energy budget method proposed by Kustas et al.(1994). In this degree-day algorithm the increased melting later in the season is simulated using a net radiation term.

Melt = tfactor $*t_{max}$ + nfactor $*Q_nD$, where the observation Q_nD is the net all-wave radiation calculated from a module.

tfactor and nfactor equal to zero

If both parameters tfactor and nfactor are set equal to zero for an HRU, the module uses the original ebsm.f algorithm otherwise the simplified degree-day algorithm is used using the values set in tfactor and nfactor.

Observation QnD is used if available, otherwise Qn is calculated as in the original algorithm.

Over snow during melt;

Qn = -0.53+0.47*(QdroD+QdfoD)*(0.52+0.52*sunact/sunmax)*(1.0-Albedo)

An external observation QnD may be generated using modules long, NeedleLeaf

and macro modules Slope_Qsi.mcr and Convert_Qsn_QnD.mcr.

MSM.

This module defined in ClassMSM is the minimal snow model.

Observations

- t (°C) air temperature.
- rh (%) relative humidity.
- u (m/s) average wind speed over time step.
- Qsi (W/m²) incident short-wave.
- Qli (W/m^2) incident long-wave.

Variables

- alb () snow albedo.
- snowmelt (kg/m^2*int) snowmelt.
- cumsnowmelt (kg/m²) cummulative snowmelt.
- snowmeltD (kg/m2*d) daily snowmelt.
- sursubl (kg/m^2*int) surface sublimation.
- T0 (°C) surface temperature.
- sursubl (kg/m²) cummulative surface sublimation.
- LE (W/m^2) latent heat flux .
- H (W/m^2) sensible heat flux.
- Hsm (W/m^2) snowmelt heat flux.
- LWn (W/m²) net long-wave radiation.
- SWn (W/m^2) net short-wave radiation.
- Pa (kPa) atmospheric pressure calculated from the elevation.

Parameters

- a1 (s) Albedo decay time constant for cold snow.
- a2 (s) Albedo decay time constant for melting snow.
- amin () Minimum albedo for aged snow.
- amax () Maximum albedo for fresh snow.
- smin (mm) Minimum snowfall to refresh snow albedo.
- Z0snow (m) snow roughnessength.
- Zref (m) reference height.

- basin_area (km^2) basin area.
- hru area (km^2) HRU area.
- hru_elev (m) altitude.

Variable Inputs

- SWE (*) (mm).
- net_snow (*) (mm).

snobal

This module is defined in Classsnobal.

snobal is an interactive point model using the energy balance to calculate snowmelt, and to predict runoff, from input data on snow properties, measurement heights & depths, and energy exchanges. Similar to the approach used by *Anderson (1976)*, and *Morris (1982)*, but designed to run on simpler, more generalizable inputs. The model was first presented by *Marks (1988)*, described conceptually by *Marks, et. al (1992)* and *Marks and Dozier (1992)*, and then described in great detail by *Marks, et. al (1997)*.

The model approximates the snow cover as being composed of two layers, a surface fixed-thickness active layer and a lower layer, solving for the temperature (C) and specific mass (kg/m^2) or mass per unit area (from density * depth (kg/m^3 * m)) for each, and computing total snowcover temperature and specific mass from these.

Melt is computed in either layer when the accumulated energy exceeds the "cold content" or when the "cold content" is > 0.0. Cold content is the energy required to bring the snow cover temperature up to freezing (0 C). Runoff is estimated when the accumulated melt and liquid H2O content exceed a specified threshold.

Trouble with the routine "hle1" iterative solution not converging. After ITMAX iterations an attempt is made to solve using a simplified stability function. This is tested by constraining the sensible heat to < 600 (W/m^2) and the latent heat to > -300. If either is outside the range, both are set to zero. The simple stability function can be selected instead of the iterative solution with the parameter "M_O_iter".

The simple stability function is $Ustar^2 t_a/(kgz_a(t_a-t_s))$, where k = 0.41, g = 9.81 m/s², z_a is the instrument height and t_a and t_s are the air and surface temperatures respectively.

If the snowfall is less than 1 (kg/m^2/s), it is immediately melted. This amount is accumulated and printed in the log screen as "melt_direct_cum".

To minimise the amount of snowfall immediately being melted because it was

less then the 1 mm threshold, the parameter option "ppt_daily_distrib" was added to the module "obs" to dump all of the daily snowfall into the first interval of the day instead of distributing it evenly over the day.

Observations

• none

<u>Variables</u>

- layer_count () number of layers in snowcover.
- isothermal () melting: 0/1
- snowcover () snow on ground at start of current timestep: 0/1
- R_n (W/m^2) net allwave radiation.
- H (W/m^2) sensible heat transfer.
- L_v_E (W/m^2) latent heat transfer.
- G (W/m^2) heat transfer by conduction & diffusion from soil to snowcover.
- M (W/m²) advected heat from precip.
- delta_Q (W/m^2) change in snowcover's energy.
- G_0 (W/m^2) transfer by conduction & diffusion from soil or lower layer to active layer.
- delta_Q_0 (W/m^2) change in active layer's energy.
- cc_s (J/m^2) snowcover's cold content.
- cc_s_0 (J/m^2) active layer cold content.
- cc_s_l (J/m^2) lower layer cold content.
- E_s_int (kg/m^2*int) mass of evap into air & soil from snowcover.
- E_int (kg/m²*int) mass flux by evap into air from active layer.
- melt_int (kg/m^2*int) specific melt (kg/m^2 or mm.
- snowmelt_int (kg/m^2*int) snow melt at bottom of pack.
- snowmeltD 9mm/d) daily snow melt at bottom of pack.
- cumsnowmelt (mm) cumulative snow melt at bottom of pack.
- z_s (m) total snowcover thickness.
- z_s_0 (m) active layer depth.
- z_s_l (m) lower layer depth.
- rho (kg/m³) average snowcover density.
- SWE (kg/m²) snowcover's specific mass.
- m_s_0 (kg/m^2) active layer specific mass.

- m_s_l (kg/m^2) lower layer specific mass.
- T_s (°C) average snowcover temp.
- T_s_0 (°C) active layer temp.
- T_s _l (°C) lower layer temp.
- h2o_sat () % of liquid H2O saturation (relative water).
- h2o_vol () liquid h2o content as volume ratio: V_water/(V_snow V_ice).
- h2o_max (kg/m^2) max liquid h2o content as specific mass.
- z_snow (m) depth of snow in precip.
- h2o_sat_snow () snowfall's % of liquid H2O saturation.
- precip_now () precipitation in current timestep 0/1.
- T_rain (°C) temperature of rain.
- T_rain (°C) temperature of snowfall.

Parameters

- relative_hts () measurements heights, z_T and z_u, are relative to snow 0/1.
- z_g (m) depth of soil temperature measurement.
- z_u (m) height of wind measurement.
- z_T (m) height of air temperature & vapour pressure measurements.
- z_0 (m) roughness length.
- max_z_s_0 (m) -maximum active layer thickness.
- max_h2o_vol () max liquid h2o content as volume ratio: V_water/(V_snow - V_ice).
- basin_area (km²) basin area.
- hru area (km²) HRU area.
- hru_elev (m) altitude.
- M_O_iter () Monin-Obukhov stability fuction. 0/1 simplified/ original iterated solution.

Variable Inputs

- hru_S_n (W/m^2) net solar radiation.
- hru_I_lw (W/m^2) incoming longwave (thermal) radiation.
- hru_t (°C) air temperature at height z_T.
- hru_ea (kPa) vapour pressure at height z_T.
- hru_u (m/s) wind speed at height z_u.
- hru_T_g(°C) soil temp at depth z_g.

- hru_T_pp(°C) temperature of precipitation.
- net_p (mm/int) net precipitation below canopy.
- net_rain (mm/int) net rain below canopy.
- net_snow (mm/int) net snow below canopy.
- hru_drift (mm/int) specific mass of drifting snow.
- hru_subl(mm/int) specific mass of drifting snow.
- hru_rho_snow (kg/m^3) density of snowfall.

Notes

NAME

g_soil - conduction heat flow between snow and soil

SYNOPSIS

double g_	soil(
doub	le rho,	/*	snow layer's density (kg/m^3)	*/
doub	le tsno,	/*	snow layer's temperature (K)	*/
doub	le tg,	/*	soil temperature (K)	*/
doub	le ds,	/*	snow layer's thickness (m)	*/
doub	le dg,	/*	dpeth of soil temperature measurement (m)	*/
doub	le pa)	/*	air pressure (Pa)	*/

DESCRIPTION

g_soil calculates the heat flow between a snow layer and the soil accounting for both conduction and vapor transport. See pages 45 - 47 in the reference below.

RETURN VALUE

heat transfer between soil and snow (J/m^2)

HISTORY

Aug 1984 written by D. Marks, CSL (GSFC), UCSB; May 1995 Converted from QDIPS to IPW by J. Domingo, OSU

SEE ALSO

Anderson 1976

NAME

g_snow - conduction heat flow between two snow layers

SYNOPSIS

double g_snow(

double	rho1,	/* up	oper	snow	layer's	density	(kg/m^3)	*/
double	rho2,	/* lo	ower	"	"	п	(kg/m^3)	*/
double	ts1,	/* up	oper	snow	layer's	temperat	ure (K)	*/
double	ts2,	/* lo	ower	п	11	"	(K)	*/
double	ds1,	/* up	oper	snow	layer's	thicknes	ss (m)	*/
double	ds2,	/* lo	ower	п	11	п	(m)	*/
double	pa)	/* a:	ir pr	essur	re (Pa)			*/

DESCRIPTION

g_snow calculates the heat flow between two snow layers for both conduction and vapor transport. See pages 46 and 47 of the reference below.

RETURN VALUE

heat transfer between snow layers (J/m^2)

HISTORY

Aug 1986 written by D. Marks, CSL, UCSB; May 1995 Converted to IPW by J. Domingo, OSU

SEE ALSO

Anderson 1976

NAME

hle1 - sensible and latent heat from data at 1 height

SYNOPSIS

int hle1(

double	press,	/*	air pressure (Pa)	*/
double	ta,	/*	air temperature (K) at height za	*/
double	ts,	/*	surface temperature (K)	*/
double	za,	/*	height of air temp measurement (m)	*/
double	ea,	/*	vapor pressure (Pa) at height zq	*/
double	es,	/*	vapor pressure (Pa) at surface	*/
double	zq,	/*	height of spec hum measurement (m)	*/
double	u,	/*	wind speed (m/s) at height zu	*/
double	zu,	/*	height of wind speed measurement (m)	*/
double	z0,	/*	roughness length (m)	*/

```
/* output variables */
          *h,
                /* sens heat flux (+ to surf) (W/m^2)
  double
  double
          *le, /* latent heat flux (+ to surf) (W/m^2) */
```

/* mass flux (+ to surf) (kg/m^2/s) */ double *e)

*/

DESCRIPTION

hle1 computes sensible and latent heat flux and mass flux given measurements of temperature and specific humidity at surface and one height, wind speed at one height, and roughness length. The temperature, humidity, and wind speed measurements need not all be at the same height.

RETURN VALUE

- successful calculation 0 no convergence -1
- bad input -2

HISTORY

Jun 1987 Written as a Qdips program by J. Dozier, CRSEO, UCSB; Oct 1990 Translated into IPW by K. Longley, OSU, ERLC;

SEE ALSO

Brutsaert 1982

CRHM implementation.

Trouble experienced with iterative solution not converging. Added simplified non-iterative solution. Better, but still blowing up. Added check to return h = 0.0 and le = 0.0, if H > 600 or Le < -300. This allows the model to complete the run. Values of h, le,t, ts, u, ea and es are output to allow for further investigation. _____ _

snobalCRHM

This module is defined in ClasssnobalCRHM.

snobal is an interactive point model using the energy balance to calculate snowmelt, and to predict runoff, from input data on snow properties, measurement heights & depths, and energy exchanges. Similar to the approach used by *Anderson (1976)*, and *Morris (1982)*, but designed to run on simpler, more generalizable inputs. The model was first presented by *Marks (1988)*, described conceptually by *Marks, et. al (1992)* and *Marks and Dozier (1992)*, and then described in great detail by *Marks, et. al (1997)*.

The model approximates the snow cover as being composed of two layers, a surface fixed-thickness active layer and a lower layer, solving for the temperature (C) and specific mass (kg/m^2) or mass per unit area (from density * depth (kg/m^3 * m)) for each, and computing total snowcover temperature and specific mass from these.

Melt is computed in either layer when the accumulated energy exceeds the "cold content" or when the "cold content" is > 0.0. Cold content is the energy required to bring the snow cover temperature up to freezing (0 C). Runoff is estimated when the accumulated melt and liquid H2O content exceed a specified threshold.

Trouble with the routine "hle1" iterative solution not converging. After ITMAX iterations an attempt is made to solve using a simplified stability function. This is tested by constraining the sensible heat to < 600 (W/m^2) and the latent heat to > -300. If either is outside the range, both are set to zero. The simple stability function can be selected instead of the iterative solution with the parameter "M_O_iter".

The simple stability function is $Ustar^2 t_a/(kgz_a(t_a-t_s))$, where k = 0.41, g = 9.81 m/s², z_a is the instrument height and t_a and t_s are the air and surface temperatures respectively.

If the snowfall is less than 1 (kg/m^2/s), it is immediately melted. This amount is accumulated and printed in the log screen as "melt_direct_cum".

To minimise the amount of snowfall immediately being melted because it was

less then the 1 mm threshold, the parameter option "ppt_daily_distrib" was added to the module "obs" to dump all of the daily snowfall into the first interval of the day instead of distributing it evenly over the day.

Observations

- Qsi (W/m^2) incoming solar radiation
- Qli (W/m^2) incoming longwave (thermal) radiation.

Variables

- layer_count () number of layers in snowcover.
- isothermal () melting: 0/1
- snowcover () snow on ground at start of current timestep: 0/1
- R_n (W/m^2) net allwave radiation.
- H (W/m^2) sensible heat transfer.
- L_v_E (W/m^2) latent heat transfer.
- G (W/m^2) heat transfer by conduction & diffusion from soil to snowcover.
- M (W/m²) advected heat from precip.
- delta_Q (W/m^2) change in snowcover's energy.
- G_0 (W/m^2) transfer by conduction & diffusion from soil or lower layer to active layer.
- delta_Q_0 (W/m^2) change in active layer's energy.
- cc_s (J/m^2) snowcover's cold content.
- cc_s_0 (J/m^2) active layer cold content.
- cc_s_l (J/m^2) lower layer cold content.
- E_s_int (kg/m^2*int) mass of evap into air & soil from snowcover.
- E_int (kg/m²*int) mass flux by evap into air from active layer.
- melt_int (kg/m^2*int) specific melt (kg/m^2 or mm.
- snowmelt_int (kg/m^2*int) snow melt at bottom of pack.
- snowmeltD 9mm/d) daily snow melt at bottom of pack.
- cumsnowmelt (mm) cumulative snow melt at bottom of pack.
- z_s (m) total snowcover thickness.
- z_s_0 (m) active layer depth.
- z_s_l (m) lower layer depth.
- rho (kg/m³) average snowcover density.
- SWE (kg/m²) snowcover's specific mass.

- m_s_0 (kg/m^2) active layer specific mass.
- m_s_l (kg/m^2) lower layer specific mass.
- T_s (°C) average snowcover temp.
- T_s_0 (°C) active layer temp.
- T_s _l (°C) lower layer temp.
- h2o_sat () % of liquid H2O saturation (relative water).
- h2o_vol () liquid h2o content as volume ratio: V_water/(V_snow V_ice).
- h2o_max (kg/m^2) max liquid h2o content as specific mass.
- z_snow (m) depth of snow in precip.
- h2o_sat_snow () snowfall's % of liquid H2O saturation.
- precip_now () precipitation in current timestep 0/1.
- T_rain (°C) temperature of rain.
- T_rain (°C) temperature of snowfall.

Parameters

- relative_hts () measurements heights, z_T and z_u, are relative to snow 0/1.
- z_g (m) depth of soil temperature measurement.
- hru_T_g (°C) assume constant temperature at depth z_g during melt.
- z_u (m) height of wind measurement.
- z_T (m) height of air temperature & vapour pressure measurements.
- z_0 (m) roughness length.
- max_z_s_0 (m) -maximum active layer thickness.
- max_h2o_vol () max liquid h2o content as volume ratio: V_water/(V_snow - V_ice).
- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- hru_elev (m) altitude.
- M_O_iter () Monin-Obukhov stability fuction. 0/1 simplified/ original iterated solution.
- hru_rho_snow (kg/m^3) density of snowfall.
- rain_soil_snow () 0 handle only snow (infiltration routine handles rain),
 1 handle snow and rain (rain is added to the snowpack)..

Variable Inputs

• albedo () - snow surface albedo.

- hru_t (°C) air temperature at height z_T.
- hru_ea (kPa) vapour pressure at height z_T.
- hru_u (m/s) wind speed at height z_u.
- hru_T_pp(°C) precipitation temperature assumed to be the air temperature.
- net_p (mm/int) net precipitation below canopy.
- net_rain (mm/int) net rain below canopy.
- net_snow (mm/int) net snow below canopy.
- hru_drift (mm/int) specific mass of drifting snow.
- hru_subl(mm/int) specific mass of drifting snow.

Notes

NAME

g_soil - conduction heat flow between snow and soil

SYNOPSIS

```
double g_soil(
    double rho, /* snow layer's density (kg/m^3)
                                                               */
                   /* snow layer's temperature (K)
                                                               */
    double tsno,
    double tg,
                   /* soil temperature (K)
                                                               */
                   /* snow layer's thickness (m)
    double ds,
                                                               */
    double dg,
                   /* dpeth of soil temperature measurement (m)
                                                               */
    double pa)
                   /* air pressure (Pa)
                                                               */
```

DESCRIPTION

g_soil calculates the heat flow between a snow layer and the soil accounting for both conduction and vapor transport. See pages 45 - 47 in the reference below.

RETURN VALUE

heat transfer between soil and snow (J/m^2)

HISTORY

Aug 1984 written by D. Marks, CSL (GSFC), UCSB; May 1995 Converted from QDIPS to IPW by J. Domingo, OSU

SEE ALSO

Anderson 1976

NAME

g_snow - conduction heat flow between two snow layers

SYNOPSIS

```
double g_snow(
    double rho1, /* upper snow layer's density (kg/m^3)
                                                    */
    double rho2, /* lower " " (kg/m^3)
                                                    */
    double ts1, /* upper snow layer's temperature (K)
                                                    */
    double ts2, /* lower " "
                                       п
                                              (K)
                                                    */
    double ds1, /* upper snow layer's thickness (m)
                                                    */
    double ds2, /* lower " "
                                       п
                                             (m)
                                                    */
    double pa) /* air pressure (Pa)
                                                    */
```

DESCRIPTION

g_snow calculates the heat flow between two snow layers for both conduction and vapor transport. See pages 46 and 47 of the reference below.

RETURN VALUE

heat transfer between snow layers (J/m^2)

HISTORY

Aug 1986 written by D. Marks, CSL, UCSB; May 1995 Converted to IPW by J. Domingo, OSU

SEE ALSO

Anderson 1976

NAME

hle1 - sensible and latent heat from data at 1 height

SYNOPSIS

int hle1(

double	press,	/*	air pressure (Pa)	*/
double	ta,	/*	air temperature (K) at height za	*/
double	ts,	/*	surface temperature (K)	*/
double	za,	/*	height of air temp measurement (m)	*/
double	ea,	/*	vapor pressure (Pa) at height zq	*/
double	es,	/*	vapor pressure (Pa) at surface	*/
double	zq,	/*	height of spec hum measurement (m)	*/
double	u,	/*	wind speed (m/s) at height zu	*/

```
zu, /* height of wind speed measurement (m) */
  double
                  /* roughness length (m)
                                                          */
  double
           z0,
/* output variables */
                /* sens heat flux (+ to surf) (W/m^2)
  double
          *h,
                                                          */
                  /* latent heat flux (+ to surf) (W/m^2) */
          *le,
  double
          *e) /* mass flux (+ to surf) (kg/m^2/s)
                                                          */
  double
```

DESCRIPTION

hle1 computes sensible and latent heat flux and mass flux given measurements of temperature and specific humidity at surface and one height, wind speed at one height, and roughness length. The temperature, humidity, and wind speed measurements need not all be at the same height.

RETURN VALUE

Θ	successful	calculation
Θ	successtul	calculation

- -1 no convergence
- -2 bad input

HISTORY

Jun 1987

Written as a Qdips program by J. Dozier, CRSEO, UCSB;

Oct 1990

Translated into IPW by K. Longley, OSU, ERLC;

SEE ALSO

Brutsaert 1982

CRHM implementation.

Trouble experienced with iterative solution not converging. Added simplified non-iterative solution. Better, but still blowing up. Added check to return h =

0.0 and le = 0.0, if H > 600 or Le < -300. This allows the model to complete the run. Values of h, le,t, ts, u, ea and es are output to allow for further investigation.

XG

This module is defined in ClassXG and is a simple thaw-freeze algorithm for a multi-layered soil using the Stephan Equation. This implementation is from Xie Chengwei and William A. Gough, 2013. The module is implemented as follows,

- 1. uses the ground surface temperature (hru_tsf).
- 2. variation #1 uses air temperature (hru_t) and handles the thermal coductivity of the snowpack.
- 3. variation #2 uses air temperature (hru_t) and the parameter n_factor to estimate the ground surface temperature under the snowpack.

Observations

• none.

Variables

- Zdf (m) depth of surface freezing front.
- Zdt (m) depth of surface thawing front.
- Zd_front (m) all freezing/thawing fronts. Allowable number of fronts determined by nlay. Thaw fronts are positive depths, freeze fronts are negative depths. Variable Zd_front(HRU, 1) always equals current (-Zdf) or Zdt.
- nfront () number of freezing/thawing fronts. Maximum number of fronts equals number of layers (nlay) minus one.
- Bfr (°C*d) freeze degree days.
- Bth ((°C*d) thaw degree days.
- pf () soil layers freezing ratios.
- pt () soil layers thawng ratios
- ttc (W/(m*K) soil layers thawing thermal coductivity.
- ftc (W/(m*K) soil layers freezing thermal coductivity.
- theta (m³/m³) XG layer theta calulated from soil moistures and defaults.
- layer_h2o (kg/m^3) layer water content.
- B (°C*d) used to sum interval values to create degree days. Local.
- TrigAcc (°C*d) detects freeze/thaw cycle. Local

- TrigState () 1/0/-1 -> thaw/idle/freeze. Local.
- SWE_tc (W/(m*K) SWE thermal coductivity. (only variation #1)
- SWE_density (kg/m³) average density of SWE. (only variation #1)
- SWE_Bfr (°C*d) thaw degree days for snowpack. (only variation #1)
- SWE_Za (m) physical depth of snowpack. (only variation #1)
- ttc_contents () ttc value when thawing 0/1 thaw/freeze.
- ftc_contents () ftc value when freezing 0/1 freeze/thaw.
- tc_composite (W/(m*K) freeze/thaw composite ftc/ttc value.
- tc_composite2 (W/(m*K) freeze/thaw composite ftc/ttc value.
- XG_moist_d (m) layer depth of soilX soil moisture in XG.
- XG_rechr_d (m) layer depth of soilX recharge in XG.
- XG_max (mm) layer maximum soil moisture content.
- XG_moist (mm) layer moisture comtent.
- check_XG_moist (mm) sum of XG layer soil moistures, should equal SoilX soil_moist value.
- rechr_fract () fraction of layer calculated from soilX soil_rechr_max. Used by to distribute soilX soil_rechr through layers.
- moist_fract () fraction of layer calculated from soilX soil_rechr_max. Used by to distribute soilX soil_moist through layers.
- default_fract () fraction of layer not affected by XG soil_moist and defaults to value of "theta_default".
- n_factor () calculated value of n_factor from parameters. (variation #2)

Parameters

- Trigthrhld (°C*d) Trigger reference level in degree days.
- depths (m) thicknesses of soil layers.
- por () soil porosity.
- N_Soil_layers () number of soil layers. Must be less than or equal nlay. Define parameters for these layers. All others are ignored.
- theta_default (m^3/m^3) value of theta used where soil moisture is not nassigned by module soilX.
- theta_min (m^3/m^3) minimum value of theta to prevent divide by zero errors.
- soil_type () HRU soil type: 0 = peat, 1 = silt, 2 = silt2, 3 = Sandy, 4 = gravel.
- calc_coductivity () 0 linear method, 1 use Johansen 1975.
- soil_solid_km (W/(m*K) dry soil thermal conductivity. Suggested values for mineral and loam are 2.5 and 0.06.

- soil_solid_km_ki (W/(m*K) saturated frozen soil thermal conductivity. Suggested values for mineral and loam are 2.5 and 0.06.
- soil_solid_km_km (W/(m*K) saturated unfrozen soil thermal conductivity. Suggested values for mineral and loam are 2.5 and 0.06.
- n_factor_a () constant ratio of surface to air temperature. (variation #2)
- n_factor_b () constant ratio of surface to air temperature. (variation #2)
- n_factor_c () constant ratio of surface to air temperature. (variation #2)
- n_factor_T () constant ratio of surface to air temperature. (variation #2)
- Zdf_init (m) initial depth of freezing front.
- Zdt_init (m) initial depth of thawing front.
- Zdp_init (m) initial depth of permafrost.
- freeze_kw_ki_update () Change thermal conductivities kw to ki behind freezing front.
- thaw_ki_kw_update () Change thermal conductivities ki to kw behind freezing front.
- k_update () Change thermal conductivities ki to kw 0 never, 1 _ after processing layer, 2 continuously.

Variable Inputs

- hru_tsf (°C) surface temperature in basic module. Basic module.
- hru_t (°C) air temperature used with SWE in variations #1 and with modifiers n_factors in variations #2.
- SWE (kg/m²) Snow water equivalent.((variation #1 and variation #2)).

Notes.

ftc_lay = (1.0 - por_lay)*km + h2o_lay*kw + (por_lay - h2o_lay)*ka
ttc_lay = (1.0 - por_lay)*km + h2o_lay*ki + (por_lay - h2o_lay)*ka
where;

representative values are, const float ko = 0.21; // W/(m K) organic material const float km = 2.50; // W/(m K) mineral const float ka = 0.025; // W/(m K) air const float ki = 2.24; // W/(m K) ice const float kw = 0.57; // W/(m K) water

```
when SWE_density[hh] > 0.156
    SWE_tc = 0.138 - 1.01*SWE_density/1000.0 +
3.233*sqr(SWE_density/1000.0)
else
    SWE_tc = 0.023 + 0.234*SWE_density/1000.0
```

In variation#1 when snowcover is handled some assumptions are made in order to avoid handling the internal energy of the snowpack.

- 1. Freezing, the snowpack is never allowed to diminish in depth as this is caused by interaction with the atmosphere by other processes rather than sensible heat transfer. Early on in the season when the ground is freezing, the snowpack is shallow and the ground surface is close to zero. At melt there is little more freezing at depth and again the temperature of the ground surface is close to zero. As a result the influence of ground temperature during the periods when the snowpack is not being handled correctly is small.
- 2. Thawing, the majority of ground thawing occurs after the snowcover has disappeared and the snowpack can ignored.

Melt Routine Features.

This description of melt routine features will help to clarify the differences in operation of the modules, *ebsm*, *MSM* and *Kevin*.

Temperature Index and Simplified Energy Budget Method.

If either tfactor or nfactor is non zero this mode is selected.

Temperature Index.

Units of tfactor are (mm/d*°C).

- 1. *ebsm* Daily calculation. Melt = tmax[0]*tfactor (mm/day). Uses daily maximum temperature.
- Kevin Interval calculation. Melt = hru_t[hh]*tfactor/Freq (mm/int). Is set to zero unless hru_t is > 0°C. Where Freq is the number of intervals in a day.

Simplified Energy Budget Method proposed by Kustas et al. (1994).

Units of nfactor are mm/(MJ/m^2)/day. This converts the energy input to mm (kg/m2) of melt.

- 1. *ebsm* Daily calculation. Melt = QnD*nfactor (mm/day).
- 2. *Kevin* Interval calculation. Melt = hru_Qn *WtoMJ*nfactor/Freq (mm ∆t).

nfactor converts energy to melt. It is the latent heat of fusion, 0.3336 MJ/kg. However, it may be adjusted by the thermal quality B = 0.95.

Simplified Energy Budget Sources.

- 1. QnD (W/m²) may be calculated using modules *slope_Qsi*, *long*, *albedo*, *NeedleLeaf* and *Convert_Qnsn_QnD*.
- 2. hru_Qn (MJ/d) may be calculated using modules *slope_Qsi*, *long*, *albedo*, *NeedleLeaf* and Convert_Qnsn_hru_Qn.

Adjusting for Cloud Cover.

Hours of bright sunshine used to be measured at meteorology stations. Today that is rarely done. Earlier algorithms for calculating short wave and long wave radiation used the ratio of actual sunshine hours to the maximum possible in the day (n/N). The module ebsm uses this ratio. When the sunshine hours are not measured the module *calcsun* can estimate (n/N) from the measured incoming short-wave Qsi and the clear sky Qsi calculated by the module *global*.

Adjusting Calculated Incoming Short-wave on a Slope for Cloud Cover.

Since radiation measurements are not normally made on a slope, it is necessary to calculated the incoming short-wave using the module global and then reducing the value for cloud cover. This is done by adjusting the value using the ratio of Qsi measured on the level divided by the calculated Qsi for the level. The module MSM uses these inputs.

ebsm (tfactor and nfactor both zero).

$$Qm = Qsn + Qnl + Qh + Qe + Qg + Qp - du/dt$$

During melt Qn = Qsn + Qsl is a linear function of the daily net short-wave radiation.

Qn = -0.53+0.47*(QdroD + QdfoD)*(0.52+0.52*sunact/sunmax)*(1.0-Albedo).

MSM (tfactor and nfactor both zero).

to be completed.

Kevin (tfactor and nfactor both zero).

to be completed.

Infiltration module summary.

	Winter Infiltration (frozen ground).		Summer and Winter (unfrozen ground) Infiltration.			
	Crack	Frozen	Everything	Ayers	GreenAmpt	
Classcrack ("crack")	*		*			
ClassPrairieInfil ("PrairieInfiltration")	*			*		
ClassAyers ("Ayers")				Summer and Winter (unfrozen)		
ClassFrozenAyers ("FrozenAyers")		*		*		
ClassGreenAmpt ("GreenAmpt")					Summer and Winter (unfrozen)	
ClassGreencrack ("Greencrack")	*				*	
Classfrozen ("frozen")		*	*			

Comments.

ClassAyers ("Ayers") for unfrozen soil.

Attempts to infiltrate snowmelt before handling rainfall.

ClassGreenAmpt ("GreenAmpt") for unfrozen soil.

This module immediately combines snowmelt and net_rain before using the GreenAmpt method. It is not recommended.

Classcrack ("crack") and Classfrozen ("frozen") for frozen soil.

These module were developed for the Canadian prairies. During the summer it is assumed that everything infiltrates.

ClassPrairieInfil ("PrairieInfiltration") and ClassAyers ("FrozenAyers") for frozen soil.

These modules can limit summer infiltration using soil texture/groundcover and maximum permissible infitration rates.

ClassGreencrack ("Greencrack") for frozen soil.

GreenAmpt controls summer infiltration and is not recommended.

GreenAmpt.

This module is defined in ClassGreenAmpt. It handles unfrozen soil infiltration.

Observations

• none

Variables

- infil (mm/int) depth of infiltration in time step, ∆t expressed as an average depth of water on an HRU.
- cuminfil (mm) cumulative infiltration expressed as an average depth of water on an HRU.
- runoff (mm/int) amount of daily runoff expressed as an equivalent depth (m³/m²).
- cumrunoff (mm) cummulative runoff.
- meltrunoff (mm/int) amount of interval melt runoff. .
- cummeltrunoff (mm) cumulative interval melt runoff.
- snowinfil (mm/int) amount of interval infiltration expressed as an equivalent depth (m³/m²).
- cumsnowinfil (mm) cumulative interval infiltration

Parameters

- basin_area (km^2) area of watershed.
- hru_area(km²) area of HRUs.
- soiltype () 0 through 12 for water/sand/loamsand/sandloam/loam/siltloam/sasclloam/clayloam/siclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/siltloam/sasclloam/siltloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/siltloam/siltloam/sasclloam/siltloam/sil
- soil_moist_max (mm) maximum available water holding capacity of soil profile.
- soil_moist_init (mm) initial value of available water in soil profile.

Variable Inputs

 net_rain (mm/int) (*) - depth of rain received by the ground surface in time step, ∆t. Variable available from module *intcp*, etc..

- snowmeltD (mm/d) (*)
- soil_moist (mm) from module *Soil*, *smba*l, etc.

Notes on Green-Ampt.

The module is used when unlimited infiltration is not possible.

When the surface ponding is negligible,

f (mm/hr) = $k^*(\psi^* \Delta \theta/F + 1)$, (eq. 1.)

where:

k (mm/hr) = saturated hydraulic conductivity.

 ψ (mm) = capillary suction at the wetting front.

 $\Delta \theta = (\theta s - \theta i)$. change in volumetric soil moisture content.

When ponded,

F1 (mm) =
$$k^*\Delta t + \psi^*\Delta \theta * \ln((F1 + \psi^*\Delta \theta)/(F0 + \psi^*\Delta \theta)).$$
 (eq. 2.)

where F0 (mm) = cumulative infiltation at the beginning of the time interval i.e. t,

f0 (mm/hr) = infiltration rate at the beginning of the time interval i.e. t,

F1 (mm) = cumulative infiltration at the end of the time interval i.e. $t+\Delta t$,

f1(mm/hr) = infiltration rate at the end of the time interval i.e. $t+\Delta t$,

It (mm/hr) = rainfall intensity during the interval.

The cumulative infiltration (Fp) at instant of ponding time is

 $Fp = k^* \psi^* \Delta \theta / (It - k). \quad (eq. 3.)$

The portion of the interval, after which ponding occurs is $\Delta t'$,

 $\Delta t' = (Fp - F0)/It.$ (eq. 4.)

Flow Chart

Calculate the maximum infiltration rates, f0 and f1 at the beginning and end of the time step using equation 1. using $F1 = F0+\Delta t^*It$.

Case 1: When f1 is >= rainfall intensity, all the rainfall infiltrates with F1 = $F0+\Delta t^*It$.

Case 2: When the rainfall intensity > f0, all the rainfall ponds and F is calculated using an iterative solution of equation 2.

Case 3: When the rainfall intensity \leq f0 but the rainfall intensity > f1, ponding occurs during the interval and the solution consists of solving for Fp using equation 3, then using equation 4 to solve the time into the step during which ponding occurs. Next, using equation 2 with Fp as F0 the cumulative infiltration F1 after the ponding time of (Δ t - Δ t') is calculated.

Assigned Soil Properties

All, except fcap are specified at 0% saturation from Chow pg 115.

Value for wilt is from Chow.

Values for por, fcap is from Floods in Canada

Soil curves of form $K = Ks / (A*Psi^3 + 1)$

Soil Type	psi	k(mm/h)	wilt	fcap	porosity	porosity for	air entry	pore
					for Soil	Evap	tension	size
water	{ 0.0	999.9	0.000	0.00	1.000	1.000	0.000	0.0
sand	{ 49.5	117.8	0.020	0.10	0.437	0.395	0.121	4.05
loamsand	{ 61.3	29.9	0.036	0.16	0.437	0.41	0.09	4.38
sandloam	{110.1	10.9	0.041	0.23	0.453	0.435	0.218	4.9
loam	{ 88.9	3.4	0.029	0.26	0.463	0.451	0.478	5.39
siltloam	{166.8	6.5	0.045	0.38	0.501	0.485	0.786	5.3
saclloam	{218.5	1.5	0.068	0.38	0.398	0.420	0.299	7.12
----------	---------	----------	-------	------	-------	--------------	---------------	------
clayloam	{208.8	1.0	0.155	0.39	0.464	0.476	0.63	8.52
siclloam	{273.3	1.0	0.039	0.40	0.471	0.477	0.356	7.75
sandclay	{239.0	0.6	0.110	0.41	0.430	0.426	0.153	10.4
siltclay	{292.2	0.5	0.056	0.43	0.479	0.492	0.49	10.4
clay	{316.3	0.3	0.090	0.46	0.475	0.482	0.405	11.4
	psi(mm)	k(mm/hr)	wilt	fcap	por	A-Soil moist	A-Hyd cond	
water	0.0	999.9	0.000	0.00	1.100	1.000000000	1.0000000}	
sand	49.5	117.8	0.020	0.10	0.437	0.000010000	0.0002000}	
loamsand	61.3	29.9	0.036	0.16	0.437	0.000006000	0.0001000}	
sandloam	110.1	10.9	0.041	0.23	0.453	0.000001124	0.0000200}	
loam	88.9	3.4	0.029	0.26	0.463	0.000000900	0.0000150}	
siltloam	166.8	6.5	0.045	0.38	0.501	0.000000300	0.0000050}	
saclloam	218.5	1.5	0.068	0.38	0.398	0.000000200	0.0000030}	
clayloam	208.8	1.0	0.155	0.39	0.464	0.000000150	0.0000020}	
siclloam	273.3	1.0	0.039	0.40	0.471	0.000000080	0.0000015}	
sandclay	239.0	0.6	0.110	0.41	0.430	0.000000070	0.0000010}	
siltclay	292.2	0.5	0.056	0.43	0.479	0.000000010	0.0000002}	
clay	316.3	0.3	0.090	0.46	0.475	0.000000006	$0.0000001\}$	

Ayers.

This module is defined in ClassAyers . It handles only unfrozen soil infiltration.

Observations

• none

Variables

- infil (mm/int) depth of infiltration in time step, △t expressed as an average depth of water on an HRU.
- cuminfil (mm) cumulative infiltration expressed as an average depth of water on an HRU.
- runoff (mm/int) amount of daily runoff expressed as an equivalent depth (m³/m²).
- cumrunoff (mm) cummulative runoff.
- meltrunoff (mm/int) amount of interval melt runoff. .
- cummeltrunoff (mm) cumulative interval melt runoff.
- snowinfil (mm/int) amount of interval infiltration expressed as an equivalent depth (m³/m²).
- cumsnowinfil (mm) cumulative interval infiltration
- melt(mm/int) interval melt equal to snowmeltD/Freq.
- •

Parameters

- basin_area (km^2) area of watershed.
- hru_area(km²) area of HRUs.
- texture () 1 through 4 for texture:- 1 coarse/medium over coarse, 2 medium over medium, 3 medium/fine over fine, 4 soil over shallow bedrock..
- groundcover () 1 through 6 for groundcover: 1 bare soil, 2 row crop, 3 poor pasture, 4 small grains, 5 good pasture, 6 forested..

Variable Inputs

- net_rain (mm/int) (*) amount of rain received by the ground surface in time step, ∆t. Variable available from module *intcp*, etc..
- snowmeltD (mm/d) (*)

Notes on Ayers.

textureproperties[texture] [groundcover] in mm/hour.

bare soil	row crop	poor pasture	small grains	good pasture	forested	
7.6	12.7	15.2	17.8	25.4	76.2	coarse over co
2.5	5.1	7.6	10.2	12.7	15.2	medium over
1.3	1.8	2.5	3.8	5.1	6.4	medium/fine fine
0.5	0.5	0.5	0.5	0.5	0.5	soil over shal bedrock

Greencrack.

This module is derived from two other CRHM modules, crack and Greenampt and is defined in ClassGreencrack. It handles frozen soil infiltration during the spring snow melt and during the remainder of the year uses Green-Ampt for unfrozen soil infiltration.

Observations

• none

<u>Variables</u>

- infil (mm/int) depth of infiltration in time step, △t expressed as an average depth of water on an HRU.
- cuminfil (mm) cumulative infiltration expressed as an average depth of water on an HRU.
- snowinfil (mm/int) amount of daily infiltration expressed as an equivalent depth (m³/m²).
- cumsnowinfil (mm) cummulative infiltration.
- runoff (mm/int) amount of daily runoff expressed as an equivalent depth (m³/m²).
- cumrunoff (mm) cummulative runoff.
- meltrunoff (mm/int) melt runoff.
- cummeltrunoff (mm) cummulative melt runoff.
- crackstat () infiltration status. Values: 0 not started, 1 to 6 #events, >6 *Limited* ended and 10 ice lens. State variable.
- crackon () indicates when the crack frozen soil routine is enabled for an HRU. 0 disabled/1 enabled.
- RainOnSnow (mm) cumulative rain on snow.

Parameters

- basin_area (km^2) area of watershed.
- hru_area(km²) area of HRUs.
- fallstat () fall status <= 0.0 or -1 for unlimited/cracked, >= 100.0 for restricted and < 0.0 limited < 100.0. Pore saturation as a percentage.

- major (mm) threshold for major melt. Default is 5 mm/day.
- soiltype 0 through 11 for water/sand/loamsand/sandloam/loam/siltloam/sasclloam/clayloam/siclloam/siltloam/sasclloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/sasclloam/sasclloam/siltloam/sasclloam/sasclloam/siltloam/sasclloam/sasclloam/siltloam/sascll
- soil_moist_init (mm) initial soil moisture content --- used to set F1.
- soil_moist_max (mm) maximum soil moisture content.
- PriorInfiltration () allow limited melt to infiltrate prior to major melt.

Variable Inputs

- hru_tmax (°C) from module *obs*.
- snowmeltD (mm/d) (*) from melt module, e.g. *ebsm*, etc..
- SWE (mm) (*) Snow water equivalent from module, e.g. *pbsm*, etc..
- net_rain (mm/int) (*) depth of rain received by the ground surface in time step, ∆t. Variable available from module *intcp*, *brushintcp* etc..
- soil_moist (mm) from module *smba*l, etc.

Notes on Crack.

The Division of Hydrology at the University of Saskatchewan (Granger et al. 1984; Gray et al., 1985), postulated that the infiltration potential of frozen soils may be grouped in three broad categories, namely: restricted, limited and unlimited.

Restricted - Infiltration is impeded by an impermeable layer, such as an ice lens on the soil surface or within the soil close to the surface. For all practical purposes, the amount of meltwater infiltration can be assumed to be negligible and that the melt goes directly to runoff and to evaporation.

Limited - Infiltration is governed primarily by the snow-cover water equivalent and the frozen water content of the top 30 cm. of soil.

Unlimited - A soil with a high percentage of large, air-filled macropores at the time of melt. Examples of soils having these properties are dry, heavily cracked clays and coarse, dry sands, organics and others.

Within the model there is not sufficient information to determine these classifications automatically and as a result the user is required to specify these properties for each HRU every fall and input this information to the model as a parameter. The one case the model does handle is when there is an early melt

and the subsequent re-freezing causing an ice lens to form. This will change both Limited to Restricted. Implementation of the infiltration to frozen soils routines is described below.

Definitions

1. Index = INF/SWE where INF=5(1- θ p)*SWE $^0.584$.

2. Potential = INF/6.

3. MELT_THRESHOLD = 5 mm. Minimum daily meltwater at which the melt routine is enabled. Lower meltwater levels are not counted as one of the six major melt events.

4. A major melt is a day when the amount of meltwater generated is greater than the MELT_THRESHOLD.

5. Six major daily melts are allowed before the limited infiltration category is changed to Restricted.

The Frozen Soil Infiltration routine

1. The Frozen Infiltration routine is enabled in an HRU when its SWE is greater than 50mm. Each HRU is handled independently.

2. Limited infiltration is triggered into operation by the first major melt. At this time, Index and Potential are calculated from the soil moisture (θ p) and the SWE of the snowpack.

3. The Frozen Infiltration routine is disabled in an HRU when its SWE reaches zero.

The three frozen soil categories are described below.

LIMITED

1. Only six major over-winter snowmelt events are possible before the infiltration potential becomes Restricted.

2. Meltwater amounts less than the MELT_THRESHOLD is handled as runoff

unless the parameter *PriorInfiltration* is set, then all of the melt is permitted to infiltrate into the soil. Once the MELT_THRESHOLD has been exceeded it is assumed that spring snowmelt has begun and only an amount of meltwater equal to MELT*Index will infiltrate and the remainder will be handled as runoff.

3. Index and Potential are recalculated if another major melt occurs with a greater SWE.

4. If the temperature the day following a major melt event is colder than -10°C, the category is changed from Limited to Restricted assuming an ice lens has formed.

5. When the SWE of the snowpack is zero, the model returns to the module that handles unfrozen infiltration.

UNLIMITED

1. All meltwater is allowed to infiltrate.

2. If the temperature the day after a major melt event is less than -10°C, the category remains unchanged and all the melt water still infiltrates into the soil.

3. When the SWE of the snowpack is zero, the model returns to the module that handles unfrozen infiltration.

RESTRICTED

1. No meltwater is allowed to infiltrate.

2. When the SWE of the snowpack is zero, the model returns to the module that handles unfrozen infiltration.

Rain on Snow.

The energy component of Rain on Snow was not handled when the Rain on Snow occurs before melt. All the Rain on Snow is accumulated before melt and released when the first melt occurs, i.e. snowmelt is greater than zero. This is can be seen by an inspection of RainOnSnow as it remains at zero until there is melt in the HRU. The Rain on Snow is accumulated before melt in an internal variable RainOnSnowA which is not accessible to the user.

Notes on Green-Ampt.

This portion of the module replaces the module srunoff which is used to simulate prairie infiltration when the assumption is made that unlimited infiltration is possible.

When the surface ponding is negligible,

 $f (mm/hr) = k^*(\psi^* \Delta \theta/F + 1), (eq. 1.)$

where:

k (mm/hr) = saturated hydraulic conductivity.

 ψ (mm) = capillary suction at the wetting front.

 $\Delta \theta = (\theta s - \theta i)$. change in volumetric soil moisture content.

When ponded,

F1 (mm) =
$$k^*\Delta t + \psi^*\Delta \theta * \ln((F1 + \psi^*\Delta \theta)/(F0 + \psi^*\Delta \theta)).$$
 (eq. 2.)

where F0 (mm) = cumulative infiltation at the beginning of the time interval i.e. t,

f0 (mm/hr) = infiltration rate at the beginning of the time interval i.e. t,

F1 (mm) = cumulative infiltration at the end of the time interval i.e. $t+\Delta t$,

f1(mm/hr) = infiltration rate at the end of the time interval i.e. $t+\Delta t$,

It (mm/hr) = rainfall intensity during the interval.

The cumulative infiltration (Fp) at instant of ponding time is

 $Fp = k^* \psi^* \Delta \theta / (It - k). \quad (eq. 3.)$

The portion of the interval, after which ponding occurs is $\Delta t'$,

 $\Delta t' = (Fp - F0)/It.$ (eq. 4.)

Flow Chart

Calculate the maximum infiltration rates, f0 and f1 at the beginning and end of the time step using equation 1. using $F1 = F0+\Delta t^*It$.

Case 1: When f1 is >= rainfall intensity, all the rainfall infiltrates with F1 = $F0+\Delta t^*It$.

Case 2: When the rainfall intensity > f0, all the rainfall ponds and F is calculated using an iterative solution of equation 2.

Case 3: When the rainfall intensity \leq f0 but the rainfall intensity > f1, ponding occurs during the interval and the solution consists of solving for Fp using equation 3, then using equation 4 to solve the time into the step during which ponding occurs. Next, using equation 2 with Fp as F0 the cumulative infiltration F1 after the ponding time of (Δ t - Δ t') is calculated.

Assigned Soil Properties

All, except fcap are specified at 0% saturation from Chow pg 115.

Value for wilt is from Chow.

Values for por, fcap is from Floods in Canada

Soil curves of form $K = Ks / (A*Psi^3 + 1)$

	nci(mm)	k(mm/hr)	wilt	fcan	nor	A Soil moist	A-Hyd
	psi(iiiii)	к(ппп/пп)	WIIU	ICap	por	A-3011 111015t	cond
water	0.0	999.9	0.000	0.00	1.100	1.000000000	1.0000000}
sand	49.5	117.8	0.020	0.10	0.437	0.000010000	0.0002000}
loamsand	61.3	29.9	0.036	0.16	0.437	0.000006000	0.0001000}

	110.1	10.9	0.041	0.23	0.453	0.000001124	0.0000200}
sandloam							
loam	88.9	3.4	0.029	0.26	0.463	0.000000900	$0.0000150\}$
siltloam	166.8	6.5	0.045	0.38	0.501	0.000000300	0.0000050}
saclloam	218.5	1.5	0.068	0.38	0.398	0.000000200	0.0000030}
clayloam	208.8	1.0	0.155	0.39	0.464	0.000000150	0.0000020}
siclloam	273.3	1.0	0.039	0.40	0.471	0.000000080	0.0000015}
sandclay	239.0	0.6	0.110	0.41	0.430	0.000000070	0.0000010}
siltclay	292.2	0.5	0.056	0.43	0.479	0.000000010	0.0000002}
clay	316.3	0.3	0.090	0.46	0.475	0.000000006	$0.0000001\}$

crack

(Granger et al. 1984; Gray et al., 1986)

This module is defined in Classcrack and handles frozen soil infiltration during the spring snow melt. This module is based upon 15 years of study of the snow hydrology of the Prairie region and results reported in the former USSR.

Observations

• none

<u>Variables</u>

- snowinfil (mm/d) amount of daily infiltration expressed as an equivalent depth (m³/m²).
- cumsnowinfil (mm) cummulative infiltration.
- runoff (mm/d) amount of daily runoff expressed as an equivalent depth (m³/m²).
- cumrunoff (mm) cummulative runoff.
- infil (mm/int) infiltration in time step, ∆t expressed as an average depth of water on an HRU.
- cuminfil (mm) cumulative infiltration expressed as an average depth of water on an HRU.
- crackstat () HRO infiltration status. Values: 0 not started, 1 to 6 #events, >6 *Limited* ended and 10 ice lens.
- crackon () indicates when the crack frozen soil routine is enabled for an HRU. 0 disabled/1 enabled.
- RainOnSnow (mm) cumulative rain on snow.

Parameters

- fallstat () fall status <= 0.0 or -1 for unlimited/cracked, >= 100.0 for restricted and < 0.0 limited < 100.0. Pore saturation as a percentage.
- basin_area (km2) area of watershed.
- hru_area(km2) area of HRUs.
- major (mm) threshold for major melt. Default is 5 mm/day.

• PriorInfiltration () - allow 'limited'' melt to infiltrate prior to first major melt event. 0 - No/1 - Yes.

Variable Inputs

- hru_tmax (obs)
- snowmeltD (mm/d) (*)
- SWE (mm) (*)
- net_rain (mm/int) (*)

Notes

The Division of Hydrology at the University of Saskatchewan (Granger et al. 1984; Gray et al., 1985), postulated that the infiltration potential of frozen soils may be grouped in three broad categories, namely: restricted, limited and unlimited.

Restricted - Infiltration is impeded by an impermeable layer, such as an ice lens on the soil surface or within the soil close to the surface. For all practical purposes, the amount of meltwater infiltration can be assumed to be negligible and that the melt goes directly to runoff and to evaporation.

Limited - Infiltration is governed primarily by the snow-cover water equivalent and the frozen water content of the top 30 cm. of soil.

Unlimited - A soil with a high percentage of large, air-filled macropores at the time of melt. Examples of soils having these properties are dry, heavily cracked clays and coarse, dry sands, organics and others.

Within the model there is not sufficient information to determine these classifications automatically and as a result the user is required to specify these properties for each HRU every fall and input this information to the model as a parameter. The one case the model does handle is when there is an early melt and the subsequent re-freezing causing an ice lens to form. This will change both Limited to Restricted. Implementation of the infiltration to frozen soils routines is described below.

Definitions

1. Index = INF/SWE where INF=5(1- θ p)*SWE $^0.584$.

2. Potential = INF/6.

3. MELT_THRESHOLD = 5 mm. Minimum daily meltwater at which the melt routine is enabled. Lower meltwater levels are not counted as one of the six major melt events.

4. A major melt is a day when the amount of meltwater generated is greater than the MELT_THRESHOLD.

5. Six major daily melts are allowed before the limited infiltration category is changed to Restricted.

The Frozen Soil Infiltration routine

1. The Frozen Infiltration routine is enabled in an HRU when its SWE is greater than 50mm. Each HRU is handled independently.

2. Limited infiltration is triggered into operation by the first major melt. At this time, Index and Potential are calculated from the soil moisture (θ p) and the SWE of the snowpack.

3. The Frozen Infiltration routine is disabled in an HRU when its SWE reaches zero.

The three frozen soil categories are described below.

LIMITED

1. Only six major over-winter snowmelt events are possible before the infiltration potential becomes Restricted.

2. Meltwater amounts less than the MELT_THRESHOLD is handled as runoff unless the parameter *PriorInfiltration*are is set, then all of the melt is permitted to infiltrate into the soil. Once the MELT_THRESHOLD has been exceeded it is assumed that spring snowmelt has begun and only an amount of meltwater equal to MELT*Index will infiltrate and the remainder will be handled as runoff.

3. Index and Potential are recalculated if another major melt occurs with a

greater SWE.

4. If the temperature the day following a major melt event is colder than -10°C, the category is changed from Limited to Restricted assuming an ice lens has formed.

5. When the SWE of the snowpack is zero, the model returns to the module that handles unfrozen infiltration.

UNLIMITED

1. All meltwater is allowed to infiltrate.

2. If the temperature the day after a major melt event is less than -10°C, the category remains unchanged and all the melt water still infiltrates into the soil.

3. When the SWE of the snowpack is zero, the model returns to the module that handles unfrozen infiltration.

RESTRICTED

1. No meltwater is allowed to infiltrate.

2. When the SWE of the snowpack is zero, the model returns to the module that handles unfrozen infiltration.

Rain on Snow.

The energy component of Rain on Snow was not handled when the Rain on Snow occurs before melt. All the Rain on Snow is accumulated before melt and released when the first melt occurs, i.e. snowmelt is greater than zero. This is can be seen by an inspection of RainOnSnow as it remains at zero until there is melt in the HRU. The Rain on Snow is accumulated before melt in an internal variable RainOnSnowA which is not accessible to the user.

frozen

This module defined in Classfrozen calculates the 'Limited' infiltration into soils where the infiltrability is governed by the soil moisture content (water + ice) and soil temperature at the start of snow ablation and the infiltration opportunity time.

Observations

• t0_inhibit () - optional. If its value is > 0.0 the incrementing of t0_Acc during the calibration cycle is inhibited. This input may be controlled using a macro.

Variables

- infil (mm/int) depth of infiltration in time step, △t expressed as an average depth of water on an HRU.
- cuminfil (mm) cumulative infiltration expressed as an average depth of water on an HRU.
- snowinfil (mm/int) amount of daily infiltration expressed as an equivalent depth (m³/m²).
- cumsnowinfil (mm) cummulative infiltration.
- meltrunoff (mm/int) amount of interval runoff expressed as an equivalent depth (m³/m²).
- cummeltrunoff (mm) cummulative runoff.
- runoff (mm/int) amount of interval runoff expressed as an equivalent depth (m³/m²).
- cumrunoff (mm) cummulative runoff.
- t0_Var (h) value of t0 calculated by the model.
- t0_Acc (h) infiltration opportunity time accumulator.
- Inf (mm) infiltration into a frozen soil calculated from parametric relationship.
- Julian_window () currently in Julian window handling frozen infiltration.
- Julian_lockout () lockout next frozen cycle until after this Julian date when t0_Julian is not set (i.e. 0).
- infiltype() infiltration type. PREMELT/RESTRICTED/LIMITED/UNLIMITED/SATURATED -

0/1/2/3/4 respectively.

- snowmeltD_last(mm/d) yesterday's snowmelt. Used to determine the formation of ice lens.
- SWEPk (mm) peak snow water equivalent. Used to calculate t0 when parameter t0 is equal to zero.

Parameters

- t0 (h) Infiltration opportunity time. If < 0 calculate opportunity time by running forward (not reccommended), == 0 calculate t0 from '0.65*SWE-5.0', > 0 use as opportunity time. The values are saved in t0_var for reference.
- t0_Julian () -value > 0 inhibit frozen algorithm till after this Julian date, 0
 enable frozen algorithm immediately when melt criteria is met.
- S0 (m³/m³) surface saturation.
- Si (m^3/m^3) initial soil saturation.
- C () coefficient.
- hru_tsoil (°K) soil average temperature at start of frozen infiltration.
- t_ice_lens(°C) overnight minimum to cause ice lens after major melt.
- soil_moisture_max (mm) soil moisture maximum. Common to module *smbal*.
- basin_area (km^2) area of watershed.
- hru_area(km²) area of HRUs.

Variable Inputs

- snowmeltD (*) (mm/d).
- hru_tmin (obs) (°C).
- SWE (*) (mm).
- net_rain (*) (mm/int).
- soil_moist (*) (mm).

Calculation

c*pow(S0, 2.92)*pow(1.0-Si, 1.64)*pow((273.15-tmean)/273.15, -0.45)*pow(t0, 0.44) *10

Reference

• Zhao and Gray (1999)

Notes.

- 1. Calibration. Since the opportunity time is not known until the model is run through the complete frozen infiltration period, the model has to be run twice. The user, by setting t0[HRU 1] to <= 0.0 causes the module to calculate the infiltration opportunity time. During this run the module forces all HRUs to ignore snowmelt, i.e. snow infiltration and runoff are zero. At the end of the model run the newly calculated opportunity times are transferred to the parameter and saved in t0_Var for future reference. If the run is terminated early the partially calculated values will be used. The calculated infiltration opportunity times saved in t0_Var may be manually transferred to t0 to eliminate the need for calculation in subsequent runs.
- 2. Maximum infiltration is limited to the lesser of Inf calculated from the equation above and the available storage in the module smbal, i.e. (soil_moist soil_moist_max).
- 3. Accumulation of the opportunity time when calculating. Normally 24 hours is added for every day when melt occurs. However, this may be refined by using the observation input t0_inhibit. When this input is not connected or its value is zero, the interval is added to the infiltration opportunity time. When t0_inhibit is greater than 0.0 the interval is not added.
- 4. When the average SWE is > 2.0 mm the module exits the unfrozen infiltration mode and enters the frozen infiltration cycle. While the daily snow melt (snowmeltD) is less than 2.0 mm
- 5. Rainfall precipitation is always allowed to infiltrate("infil". "runoff" is always zero.
- 6. Snowmelt occurring before snowmeltD exceeds 2.0 mm is handled as premelt. During this period, if the parameter "Si" is 1.0 the early snowmelt is handled as "meltrunoff" otherwise it is "snowinfil".
- 7. After the snowmelt is greater than 2 mm the snowmelt is handled as follows; Si >= 1.0 -> RESTRICTED, 1.0 > Si > 0.0 -> LIMITED and Si <= 0.0 -> LIMITED.
- 8. The rate of infiltration and the maximum amount of infiltration during LIMITED infiltration is controlled by the method described in 'Frozen soil infiltration Zhao and Gray (1999)'.
- 9. The Opportunity Time is calculated as the period beginning from when the snowmelt exceeds 2 mm per day until the average of all HRU SWE is less then 2 mm.

frozenAyers.

This module is defined in ClassfrozenAyers . It handles unfrozen soil infiltration using Ayers and frozen soil using Zhao and Gray (1999)..

Observations

• t0_inhibit () - optional. If its value is > 0.0 the incrementing of t0_Acc during the calibration cycle is inhibited. This input may be controlled using a macro.

<u>Variables</u>

- infil (mm/int) depth of infiltration in time step, △t expressed as an average depth of water on an HRU.
- cuminfil (mm) cumulative infiltration expressed as an average depth of water on an HRU.
- runoff (mm/int) amount of daily runoff expressed as an equivalent depth (m³/m²).
- cumrunoff (mm) cummulative runoff.
- meltrunoff (mm/int) amount of interval melt runoff. .
- cummeltrunoff (mm) cumulative interval melt runoff.
- snowinfil (mm/int) amount of interval infiltration expressed as an equivalent depth (m³/m²).
- cumsnowinfil (mm) cumulative interval infiltration
- t0_Var (h) value of t0 calculated by the model.
- t0_Acc (h) infiltration opportunity time accumulator.
- Inf (mm) infiltration into a frozen soil calculated from parametric relationship.
- Julian_window () currently in Julian window handling frozen infiltration.
- Julian_lockout () lockout next frozen cycle until after this Julian date when t0_Julian is not set (i.e. 0).
- infiltype() infiltration type. PREMELT/RESTRICTED/LIMITED/UNLIMITED/SATURATED - 0/1/2/3/4 respectively.
- snowmeltD_last (mm/d) yesterday's snowmelt. Used to determine the formation of ice lens.

• SWEPk (mm) - peak snow water equivalent. Used to calculate t0 when parameter t0 is equal to zero.

Parameters

- basin_area (km^2) area of watershed.
- hru_area(km^2) area of HRUs.
- t0 (h) Infiltration opportunity time. If < 0 calculate opportunity time by running forward (not reccommended), == 0 calculate t0 from '0.65*SWE-5.0', > 0 use as opportunity time. The values are saved in t0_var for reference.
- t0_Julian () -value > 0 inhibit frozen algorithm till after this Julian date, 0
 enable frozen algorithm immediately when melt criteria is met.
- S0 (m³/m³) surface saturation.
- Si (m^3/m^3) initial soil saturation.
- C () coefficient.
- hru_tsoil (°K) soil average temperature at start of frozen infiltration.
- t_ice_lens(°C) overnight minimum to cause ice lens after major melt.
- texture () 1 through 4 for texture:- 1 coarse/medium over coarse, 2 medium over medium, 3 medium/fine over fine, 4 soil over shallow bedrock..
- groundcover () 1 through 6 for groundcover: 1 bare soil, 2 row crop, 3 poor pasture, 4 small grains, 5 good pasture, 6 forested..
- soil_moist_max (mm) Maximum available water holding capacity of soil profile. Soil profile is surface to bottom of rooting zone.

Variable Inputs

- net_rain (mm/int) (*) amount of rain received by the ground surface in time step, ∆t. Variable available from module *intcp*, etc..
- snowmeltD (mm/d) (*).
- soil_moist (mm) (*)
- SWE (mm) (*)
- hru_tmin (°C) (*)

Notes on Ayers.

textureproperties[texture] [groundcover] in mm/hour.

bare soil	row crop	poor pasture	small grains	good pasture	forested	
7.6	12.7	15.2	17.8	25.4	76.2	coarse over co
2.5	5.1	7.6	10.2	12.7	15.2	medium over
1.3	1.8	2.5	3.8	5.1	6.4	medium/fine fine
0.5	0.5	0.5	0.5	0.5	0.5	soil over shal bedrock

Calculations used in frozen.

c*pow(S0, 2.92)*pow(1.0-Si, 1.64)*pow((273.15-tmean)/273.15, -0.45)*pow(t0, 0.44) *10

Reference

• Zhao and Gray (1999)

Notes.

- 1. Calibration. Since the opportunity time is not known until the model is run through the complete frozen infiltration period, the model has to be run twice. The user, by setting t0[HRU 1] to <= 0.0 causes the module to calculate the infiltration opportunity time. During this run the module forces all HRUs to ignore snowmelt, i.e. snow infiltration and runoff are zero. At the end of the model run the newly calculated opportunity times are transferred to the parameter and saved in t0_Var for future reference. If the run is terminated early the partially calculated values will be used. The calculated infiltration opportunity times saved in t0_Var may be manually transferred to t0 to eliminate the need for calculation in subsequent runs.
- 2. Maximum infiltration is limited to the lesser of Inf calculated from the equation above and the available storage in the module smbal, i.e. (soil_moist soil_moist_max).
- 3. Accumulation of the opportunity time when calculating. Normally 24 hours is added for every day when melt occurs. However, this may be refined by

using the observation input t0_inhibit. When this input is not connected or its value is zero, the interval is added to the infiltration opportunity time. When t0_inhibit is greater than 0.0 the interval is not added.

- 4. When the average SWE is > 2.0 mm the module exits the unfrozen infiltration mode and enters the frozen infiltration cycle. While the daily snow melt (snowmeltD) is less than 2.0 mm
- 5. Rainfall precipitation is always allowed to infiltrate("infil". "runoff" is always zero.
- 6. Snowmelt occurring before snowmeltD exceeds 2.0 mm is handled as premelt. During this period, if the parameter "Si" is 1.0 the early snowmelt is handled as "meltrunoff" otherwise it is "snowinfil".
- 7. After the snowmelt is greater than 2 mm the snowmelt is handled as follows; Si >= 1.0 -> RESTRICTED, 1.0 > Si > 0.0 -> LIMITED and Si <= 0.0 -> LIMITED.
- 8. The rate of infiltration and the maximum amount of infiltration during LIMITED infiltration is controlled by the method described in 'Frozen soil infiltration Zhao and Gray (1999)'.
- 9. The Opportunity Time is calculated as the period beginning from when the snowmelt exceeds 2 mm per day until the average of all HRU SWE is less then 2 mm.

PrairieInfiltration .

This module is defined in ClassPrairieInfiltration . It handles unfrozen soil infiltration using Ayers and frozen soil using Granger et al. 1984; Gray et al., 1986..

Observations

• none

Variables

- infil (mm/int) depth of infiltration in time step, △t expressed as an average depth of water on an HRU.
- cuminfil (mm) cumulative infiltration expressed as an average depth of water on an HRU.
- runoff (mm/int) amount of daily runoff expressed as an equivalent depth (m³/m²).
- cumrunoff (mm) cummulative runoff.
- meltrunoff (mm/int) amount of interval melt runoff. .
- cummeltrunoff (mm) cumulative interval melt runoff.
- snowinfil (mm/int) amount of interval infiltration expressed as an equivalent depth (m³/m²).
- cumsnowinfil (mm) cumulative interval infiltration
- crackstat () HRO infiltration status. Values: 0 not started, 1 to 6 #events, >6 *Limited* ended and 10 ice lens.
- crackon () indicates when the crack frozen soil routine is enabled for an HRU. 0 disabled/1 enabled.
- RainOnSnow (mm) cumulative rain on snow.

Parameters

- basin_area (km^2) area of watershed.
- hru_area(km²) area of HRUs.
- texture () 1 through 4 for texture:- 1 coarse/medium over coarse, 2 medium over medium, 3 medium/fine over fine, 4 soil over shallow bedrock..

- groundcover () 1 through 6 for groundcover: 1 bare soil, 2 row crop, 3 poor pasture, 4 small grains, 5 good pasture, 6 forested..
- fallstat () fall status <= 0.0 or -1 for unlimited/cracked, >= 100.0 for restricted and < 0.0 limited < 100.0. Pore saturation as a percentage.
- major (mm) threshold for major melt. Default is 5 mm/day.
- PriorInfiltration () allow 'limited" melt to infiltrate prior to first major melt event. 0 No/1 Yes.

Variable Inputs

- hru_tmax (obs)
- snowmeltD (mm/d) (*)
- SWE (mm) (*)
- net_rain (mm/int) (*) amount of rain received by the ground surface in time step, ∆t. Variable available from module *intcp*, etc..

Notes on Ayers.

bare soil	row crop	poor pasture	small grains	good pasture	forested	
7.6	12.7	15.2	17.8	25.4	76.2	coarse over c
2.5	5.1	7.6	10.2	12.7	15.2	medium over
1.3	1.8	2.5	3.8	5.1	6.4	medium/fine fine
0.5	0.5	0.5	0.5	0.5	0.5	soil over shal bedrock

textureproperties[texture] [groundcover] in mm/hour.

Notes on frozen routine.

The Division of Hydrology at the University of Saskatchewan (Granger et al. 1984; Gray et al., 1985), postulated that the infiltration potential of frozen soils

may be grouped in three broad categories, namely: restricted, limited and unlimited.

Restricted - Infiltration is impeded by an impermeable layer, such as an ice lens on the soil surface or within the soil close to the surface. For all practical purposes, the amount of meltwater infiltration can be assumed to be negligible and that the melt goes directly to runoff and to evaporation.

Limited - Infiltration is governed primarily by the snow-cover water equivalent and the frozen water content of the top 30 cm. of soil.

Unlimited - A soil with a high percentage of large, air-filled macropores at the time of melt. Examples of soils having these properties are dry, heavily cracked clays and coarse, dry sands, organics and others.

Within the model there is not sufficient information to determine these classifications automatically and as a result the user is required to specify these properties for each HRU every fall and input this information to the model as a parameter. The one case the model does handle is when there is an early melt and the subsequent re-freezing causing an ice lens to form. This will change both Limited to Restricted. Implementation of the infiltration to frozen soils routines is described below.

Definitions

1. Index = INF/SWE where INF=5(1- θ p)*SWE^0.584.

2. Potential = INF/6.

3. MELT_THRESHOLD = 5 mm. Minimum daily meltwater at which the melt routine is enabled. Lower meltwater levels are not counted as one of the six major melt events.

4. A major melt is a day when the amount of meltwater generated is greater than the MELT_THRESHOLD.

5. Six major daily melts are allowed before the limited infiltration category is changed to Restricted.

The Frozen Soil Infiltration routine

1. The Frozen Infiltration routine is enabled in an HRU when its SWE is greater than 50mm. Each HRU is handled independently.

2. Limited infiltration is triggered into operation by the first major melt. At this time, Index and Potential are calculated from the soil moisture (θ p) and the SWE of the snowpack.

3. The Frozen Infiltration routine is disabled in an HRU when its SWE reaches zero.

The three frozen soil categories are described below.

LIMITED

1. Only six major over-winter snowmelt events are possible before the infiltration potential becomes Restricted.

2. Meltwater amounts less than the MELT_THRESHOLD is handled as runoff unless the parameter *PriorInfiltration* is set, then all of the melt is permitted to infiltrate into the soil. Once the MELT_THRESHOLD has been exceeded it is assumed that spring snowmelt has begun and only an amount of meltwater equal to MELT*Index will infiltrate and the remainder will be handled as runoff.

3. Index and Potential are recalculated if another major melt occurs with a greater SWE.

4. If the temperature the day following a major melt event is colder than -10°C, the category is changed from Limited to Restricted assuming an ice lens has formed.

5. When the SWE of the snowpack is zero, the model returns to the module that handles unfrozen infiltration.

UNLIMITED

1. All meltwater is allowed to infiltrate.

2. If the temperature the day after a major melt event is less than -10°C, the category remains unchanged and all the melt water still infiltrates into the soil.

3. When the SWE of the snowpack is zero, the model returns to the module that handles unfrozen infiltration.

RESTRICTED

1. No meltwater is allowed to infiltrate.

2. When the SWE of the snowpack is zero, the model returns to the module that handles unfrozen infiltration.

Rain on Snow.

The energy component of Rain on Snow was not handled when the Rain on Snow occurs before melt. All the Rain on Snow is accumulated before melt and released when the first melt occurs, i.e. snowmelt is greater than zero. This is can be seen by an inspection of RainOnSnow as it remains at zero until there is melt in the HRU. The Rain on Snow is accumulated before melt in an internal variable RainOnSnowA which is not accessible to the user.

Soil

This module is defined in ClassSoil and handles soil moisture accounting throughout the year. When snow cover is present, the input to this module is the infiltration (snowinfil) generated by the modules crack, frozen etc. From the end of snow melt till late fall the infiltration (infil) is generated by the modules Greencrack, PrairieInfiltration etc. The soil is handled as two layers. The upper layer is called the recharge layer and represents the top soil. Evaporation can occur from the recharge layer, soil_moist (all layers) or none. The surface infiltration satisfies the recharge layer before being used by the lower layer. The excess water from both soil layers contribute to the ground water flow , depressional storage and then to sub surface flow. The depressional storage is allowed to drain to ground water and sub surface flow. The parameters soil_rechr_max and soil_moist_max represents the maximum soil moisture capacity for the two layers i.e. field capacity. Wilt occurs when the state variables soil_rechr and soil_moist are equal to zero.

Evaporation and subsurface runoff are handled after infiltration. The maximum amount of evapotranspiration is calculated by the module *evap*. There is a variation (#1) to handle HRU runoff through a culvert.

Observations

• none

Variables

- Sd (mm) depression storage. State variable.
- gw (mm) ground water storage. State variable.
- soil_rechr (mm) soil moisture content of recharge zone, i.e., the portion of the soil profile from which water is withdrawn by evaporation expressed as an equivalent depth (m³/m²). State variable.
- soil_moist (mm) soil moisture content of the rooting zone of the major vegetation type on the HRU - expressed as an equivalent depth (m³/m²). State variable.
- soil_ssr (mm/int) subsurface runoff from soil_rechr and depressional storage.

- cum_soil_ssr (mm) cumulative subsurface runoff from soil_rechr.
- soil_ssr_D (mm/d) daily accumulation of soil_ssr.
- rechr_ssr (mm/int) Portion of excess soil water from a HRU that enters subsurface reservoirs.
- cum_rechr_ssr (mm) Accumulation of Portion of excess from a HRU to ssr.
- soil_gw (mm/int) excess from soil_moist and depressional storage that enters groundwater reservoirs.
- cum_soil_gw (mm) cumulative portion of excess soil water from a HRU that enters groundwater reservoirs.
- soil_gw_D (mm/d) portion of excess soil water from a HRU that enters groundwater reservoirs in a day.
- soil_runoff (mm/int) remainder of excess soil water.
- cum_soil_runoff (mm) cumulative surface runoff.
- soil_runoff_D (mm/d) daily accumulation of soil_runoff.
- cum_runoff_to_Sd (mm/int) cumulative portion of runoff to depression storage.
- infil_act (mm/int) Actual amount of water infiltrating the soil on each HRU.
- infil_act_D (mm/d) Daily actual amount of water infiltrating the soil on each HRU.
- cum_infil (mm) cumulative potential amount of water infiltrating the soil on each HRU. Contents of the variable "infil" are calculated in the infiltration module used.
- cum_infil_act (mm) cumulative actual amount of water infiltrating the soil on each HRU.
- gw_flow (mm/int) Drainage from HRU ground water reservoir.
- gw_flow_D (mm/d) Daily drainage from HRU ground water reservoir.
- cum_gw_flow (mm) Accumulation of excess soil water from a HRU that enters groundwater reservoirs.
- redirected_residual (mm*km^2/int) redirected residual after topping up Sd and soil_rechar in Netroute/Netroute_D/Netroute_M/Netroute_M_D. Average over the basin.
- cum_redirected_residual (mm*km^2/int) cumulative redirected residual after topping up Sd and soil_rechar in Netroute/Netroute D/Netroute M/Netroute M D. Average over the basin.

Culvert variables only in variation #1.

- culvert_Q (m^3/s) flow in culvert.
- culvert_water_H (m) depth of pond at culvert inlet.
- culvert_water_A (m^2) surface area of culvert pond
- culvert_water_V (m^3) volume of water in culvert pond
- culvert_water_O (m³) volume of water overflowing road
- culvert_evap (mm/int) Depth of water evaporating.
- cum_culvert (mm) Cumulative culvert HRU flow.
- cum_culvert_over (mm) Cumulative culvert HRU overflow.

Parameters

- basin_area (km^2) basin area.
- hru area (km^2) HRU area.
- Sdmax (mm) Maximum depression storage.
- Sdinit (mm) Initial depression storage.
- soil_rechr_max (mm) maximum moisture content of the soil recharge zone expressed as an equivalent depth (mm³/mm²). Must be less than or equal to soil_moist.
- soil_rechr_init (mm) initial moisture content of the soil recharge zone expressed as an equivalent depth (mm³/mm²). Must be less than or equal to soil_moist_init.
- soil_moist_max (mm) maximum available water holding capacity of rooting zone expressed as an equivalent depth (mm³/mm²).
- soil_moist_init (mm) initial moisture content of rooting zone expressed as an equivalent depth (mm³/mm²).
- gw_max (mm) Maximum available water holding capacity of ground water reservoir.
- gw_init (mm) Initial value of available water in ground water reservoir.
- rechr_ssr_K (mm/d) recharge ssr drainage factor.
- lower_ssr_K (mm/d) lower column (soil_moist rechr_mois) ssr drainage factor.
- soil_gw_K (mm/d) The maximum amount of the excess soil water for an HRU that is routed directly to the associated groundwater reservoir each day- expressed as an equivalent depth (mm³/mm²).
- Sd_ssr_K (mm/d) depression storage ssr factor.
- Sd_gw_K(mm/d) depression storage groundwater factor.
- gw_K (mm/d) daily ground water drainage from gw reservoir.
- soil_withdrawal () HRU evaporation withdrawal for soil type: 1= sand,

2= loam, 3= clay. Water availability is used to limit evaporation as decribed below.

- cov_type () Vegetation cover type designation for HRU: "0 = no evaporation, 1 = bare soil or shallow crops (evaporation from the recharge layer only), 2 = crops, grasses, shrubs and trees (evaporation from all soil moisture).
- transp_limited () Inhibits transpiration from the soil moisture layer when set to 1.
- soil_ssr_runoff () soil column excess to interflow(ssr)/runoff (and possibly SD) interflow-0/runoff-1

Culvert variables only in variation #1.

- channel_slope () soil slope to culvert.
- side_slope () side soil slope mormal to culvert slope
- culvert_diam (m) culvert diameter.
- culvert_water_Dmax (m) maximum depth of pond at culvert inlet
- number_culverts () number of culverts and efficiency factor. Zero = no culvert

Variable Inputs

- hru_evap (evap) (mm/int) or (mm/d). Programmed for an interval or daily value.
- infil (*) (mm/int).
- snowinfil (*) (mm/int) or (mm/d). Programmed for an interval or daily value.
- runoff (*) (mm/int) or (mm/d). Programmed for an interval or daily value.
- meltrunoff (*) (mm/int) or (mm/d). Programmed for an interval or daily value.
- hru_actet (*) (mm/int). Programmed as a *put* value.
- hru_cum_actet (*) (mm).Programmed as a state *put* value.

Notes

Set Sdmax to zero and soil_moist to non-zero to model HRU with no Depression or Pond storage.

Set Sdmax to non-zero and soil_moist to non-zero to model HRU with

Depression storage.

Set Sdmax to non-zero and soil_moist to zero to model HRU with Pond storage.

The significant difference between Pond and Depression storage is that soil_runoff to Pond storage is unlimited to its maximum capacity whereas depression storage is also limited to (Sdmax-Sd)*(1-exp(-soil_runoff/Sdmax)).

The layer soil_rechr is part of the soil_moist zone. It is not a separate identity.

Evaporation from bare soil and shallow crops only occurs from the recharge layer. All other cover types have evaporation/transpiration from recharge and soil moisture layers.

transp_limited is provided to limit grasses, shrubs and trees to only the recharge layer when the vegetation limits transpiration during drought conditions.

infil, snowinfil and condensation are added to the recharge layer and then to soil_moist. The excess is next used to satisfy groundwater. The surplus is designated as soil_runoff. The incoming runoff and meltrunoff are added to soil_runoff. If depression storage is specified any soil_runoff is used to fill the depression storage to its maximum.

Soil Withdrawal Types.

Water availability controls evaporation withdrawal in 'smbal' and 'Soil'. Modified from Zahner(1967) by G.H.Leavesley.

1 - Sand, at pcts < 0.25, limited to 0.25*pcts*avail_evap,

2 - loam, at pcts < 0.5, limited to 0.5*pcts*avail_evap,

3 - clay, at pcts < 0.67 && pcts > 0.33, limited to pcts*avail_evap and at pcts < 0.33 limited to 0.5*pcts*avail_evap,

4 organic, unlimted availability,

where pcts = (soil moist)/(soil moist max).

SoilK

This module is defined in ClassSoilK and handles soil moisture accounting throughout the year. When snow cover is present, the input to this module is the infiltration (snowinfil) generated by the modules crack, frozen etc. From the end of snow melt till late fall the infiltration (infil) is generated by the modules Greencrack, PrairieInfiltration etc. The soil is handled as two layers. The upper layer is called the recharge layer and represents the top soil. Evaporation can occur from the recharge layer, soil_moist (all layers) or none. The surface infiltration satisfies the recharge layer before being used by the lower layer. The excess water from both soil layers contribute to the ground water flow , depressional storage and then to sub surface flow. The depressional storage is allowed to drain to ground water and sub surface flow. The parameters soil_rechr_max and soil_moist_max represents the maximum soil moisture capacity for the two layers i.e. field capacity. Wilt occurs when the state variables soil_rechr and soil_moist are equal to zero.

ClassSoilK is derived from ClassSoil with the addition of processes to handle wetlands. These processes are described in upcoming papers by Shook et Al..

There is a variation (#1) to handle HRU runoff through a culvert.

Evaporation and subsurface runoff are handled after infiltration. The maximum amount of evapotranspiration over soil is calculated by the module *evap*. Priestley-Taylor is always used over culvert ponds and wetlands.

Observations

• none

Variables

- Sd (mm) depression storage. State variable.
- gw (mm) ground water storage. State variable.
- soil_rechr (mm) soil moisture content of recharge zone, i.e., the portion of the soil profile from which water is withdrawn by evaporation expressed as an equivalent depth (m³/m²). State variable.
- soil_moist (mm) soil moisture content of the rooting zone of the major

vegetation type on the HRU - expressed as an equivalent depth (m^3/m^2). State variable.

- soil_ssr (mm/int) subsurface runoff from soil_rechr and depressional storage.
- cum_soil_ssr (mm) cumulative subsurface runoff from soil_rechr.
- soil_ssr_D (mm/d) daily accumulation of soil_ssr.
- rechr_ssr (mm/int) Portion of excess soil water from a HRU that enters subsurface reservoirs.
- cum_rechr_ssr (mm) Accumulation of Portion of excess from a HRU to ssr.
- soil_gw (mm/int) excess from soil_moist and depressional storage that enters groundwater reservoirs.
- cum_soil_gw (mm) cumulative portion of excess soil water from a HRU that enters groundwater reservoirs
- Sd_water_frac () fraction of the maximum possible Sd HRU fraction that is under water.
- contrib_frac () Fraction of the entire HRU area contributing water.
- wetlands_evap (mm/int) Depth of water evaporating from wetlands using Priestley-Taylor.
- total_evap (mm) Control variable based on running sum of evap.
- soil_gw_D (mm/d) portion of excess soil water from a HRU that enters groundwater reservoirs in a day.
- soil_runoff (mm/int) remainder of excess soil water.
- cum_soil_runoff (mm) cumulative surface runoff.
- soil_runoff_D (mm/d) daily accumulation of soil_runoff.
- cum_runoff_to_Sd (mm/int) cumulative portion of runoff to depression storage.
- infil_act (mm/int) Actual amount of water infiltrating the soil on each HRU.
- infil_act_D (mm/d) Daily actual amount of water infiltrating the soil on each HRU.
- cum_infil (mm) cumulative potential amount of water infiltrating the soil on each HRU. Contents of the variable "infil" are calculated in the infiltration module used.
- cum_infil_act (mm) cumulative actual amount of water infiltrating the soil on each HRU.
- gw_flow (mm/int) Drainage from HRU ground water reservoir.
- gw_flow_D (mm/d) Daily drainage from HRU ground water reservoir.
- cum_gw_flow (mm) Accumulation of excess soil water from a HRU that

enters groundwater reservoirs.

- redirected_residual (mm*km^2/int) redirected residual after topping up Sd and soil_rechar in Netroute/Netroute_D/Netroute_M/Netroute_M_D. Average over the basin.
- cum_redirected_residual (mm*km^2/int) cumulative redirected residual after topping up Sd and soil_rechar in Netroute/Netroute_D/Netroute_M/Netroute_M_D. Average over the basin.

Culvert variables only in variation #1.

- culvert_Q (m^3/s) flow in culvert.
- culvert_water_H (m) depth of pond at culvert inlet.
- culvert_water_A (m^2) surface area of culvert pond
- culvert_water_V (m^3) volume of water in culvert pond
- culvert_water_O (m^3) volume of water overflowing road
- culvert_evap (mm/int) Depth of water evaporating.
- cum_culvert (mm) Cumulative culvert HRU flow.
- cum_culvert_over (mm) Cumulative culvert HRU overflow.

Parameters

- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- Sdmax (mm) Maximum depression storage.
- Sdinit (mm) Initial depression storage.
- soil_rechr_max (mm) maximum moisture content of the soil recharge zone expressed as an equivalent depth (mm³/mm²). Must be less than or equal to soil_moist.
- soil_rechr_init (mm) initial moisture content of the soil recharge zone expressed as an equivalent depth (mm³/mm²). Must be less than or equal to soil_moist_init.
- soil_moist_max (mm) maximum available water holding capacity of rooting zone expressed as an equivalent depth (mm³/mm²).
- soil_moist_init (mm) initial moisture content of rooting zone expressed as an equivalent depth (mm³/mm²).
- gw_max (mm) Maximum available water holding capacity of ground water reservoir.
- gw_init (mm) Initial value of available water in ground water reservoir.

- rechr_ssr_K (mm/d) recharge ssr drainage factor.
- lower_ssr_K (mm/d) lower column (soil_moist rechr_mois) ssr drainage factor.
- soil_gw_K (mm/d) The maximum amount of the excess soil water for an HRU that is routed directly to the associated groundwater reservoir each day- expressed as an equivalent depth (mm³/mm²).
- Sd_ssr_K (mm/d) depression storage ssr factor.
- Sd_gw_K(mm/d) depression storage groundwater factor.
- gw_K (mm/d) daily ground water drainage from gw reservoir.
- Sd_water_frac_max () Maximum fraction of the entire HRU area that can be water.
- contrib_frac_init () Fraction of the entire HRU area contributing water.
- evap_threshold (mm) Threshold evaporation necessary for setting the contribution fraction to zero.
- soil_withdrawal () HRU evaporation withdrawal for soil type: 1= sand, 2= loam, 3= clay. Water availability is used to limit evaporation as decribed below.
- cov_type () Vegetation cover type designation for HRU: "0 = no evaporation, 1 = bare soil or shallow crops (evaporation from the recharge layer only), 2 = crops, grasses, shrubs and trees (evaporation from all soil moisture).
- transp_limited () Inhibits transpiration from the soil moisture layer when set to 1.
- soil_ssr_runoff () soil column excess to interflow(ssr)/runoff (and possibly SD) interflow-0/runoff-1

Culvert variables only in variation #1.

- channel_slope () soil slope to culvert.
- side_slope () side soil slope mormal to culvert slope
- culvert_diam (m) culvert diameter.
- culvert_water_Dmax (m) maximum depth of pond at culvert inlet
- number_culverts () number of culverts and efficiency factor. Zero = no culvert

Variable Inputs

• hru_evap (evap) (mm/int) or (mm/d). Programmed for an interval or daily value.
- infil (*) (mm/int).
- snowinfil (*) (mm/int) or (mm/d). Programmed for an interval or daily value.
- runoff (*) (mm/int) or (mm/d). Programmed for an interval or daily value.
- meltrunoff (*) (mm/int) or (mm/d). Programmed for an interval or daily value.
- hru_actet (*) (mm/int). Programmed as a *put* value.
- hru_cum_actet (*) (mm).Programmed as a state *put* value.

Notes

Set Sdmax to zero and soil_moist to non- zero to model HRU with no Depression or Pond storage.

Set Sdmax to non-zero and soil_moist to non- zero to model HRU with Depression storage.

Set Sdmax to non-zero and soil_moist to zero to model HRU with Pond storage.

The significant difference between Pond and Depression storage is that soil_runoff to Pond storage is unlimited to its maximum capacity whereas depression storage is also limited to (Sdmax-Sd)*(1-exp(-soil_runoff/Sdmax)).

The layer soil_rechr is part of the soil_moist zone. It is not a separate identity.

Evaporation from bare soil and shallow crops only occurs from the recharge layer. All other cover types have evaporation/transpiration from recharge and soil moisture layers.

transp_limited is provided to limit grasses, shrubs and trees to only the recharge layer when the vegetation limits transpiration during drought conditions.

infil, *snowinfil* and condensation are added to the recharge layer and then to soil_moist. The excess is next used to satisfy groundwater. The surplus is designated as soil_runoff. The incoming runoff and meltrunoff are added to soil_runoff. If depression storage is specified any soil_runoff is used to fill the

depression storage to its maximum.

Soil Withdrawal Types.

Water availability controls evaporation withdrawal in 'smbal' and 'Soil'. Modified from Zahner(1967) by G.H.Leavesley.

1 - Sand, at pcts < 0.25, limited to 0.25*pcts*avail_evap,

2 - loam, at pcts < 0.5, limited to 0.5*pcts*avail_evap,

3 - clay, at pcts < 0.67 && pcts > 0.33, limited to pcts*avail_evap and at pcts < 0.33 limited to 0.5*pcts*avail_evap,

4 organic, unlimted availability,

where pcts = (soil moist)/(soil moist max).

K_Estimate.

The rates for lateral flow rate in soil layers and groundwater layer (i.e. subsurface and groundwater discharges) as well as vertical flow of excess soil water to groundwater (i.e. groundwater recharge) are controlled by several drainage factors: rechr_ssr_K [mm day-1], lower_ssr_K [mm day-1], gw_K [mm day-1], sd_ssr_K [mm day-1], sd_gw_K [mm day-1] and soil_gw_K [mm day-1]. rechr_ssr_K, lower_ssr_K, gw_K and sd_ssr_K are the drainage factors for lateral flows in soil recharge, lower soil, groundwater layers and depressions storage, respectively. soil_gw_K and sd_gw_K are the drainage factors for the vertical flow from soil and from depression storage to groundwater layer. Darcy's law for unsaturated flow is used to calculate these drainage factors.

Observations

• none

Variables

- v_L_upper (m/s) Darcy's lateral flow velocity in upper soil column (ie. recharge layer).
- v_L_lower (m/s) Darcy's lateral flow velocity in lower soil column.
- v_V_sd (m/s) Darcy's vertical flow velocity for sd (ie. depression).
- v_V_soil (m/s) Darcy's vertical flow velocity for soil column.
- v_L_gw (m/s) -Darcy's lateral flow velocity for groundwater reservoir.

Parameters

- Ks_lower (m/s) saturated hydraulic conductivity for lower soil.
- Ks_upper (m/s) saturated hydraulic conductivity for upper soil layer.
- Ks_gw (m/s) saturated hydraulic conductivity for groundwater layer.
- hru_GSL (°) ground slope increasing the slope positively, tilts the plane to the north with ASL = 0.
- porosity (m^3/m^3) soil porosity.
- PSD () pore size distribution.
- soil_rechr_max (mm) Maximum value for soil recharge zone (upper portion of soil_moist where losses occur as both evaporation and

transpiration). Must be less than or equal to soil_moist.

- soil_moist_max (mm) Maximum available water holding capacity of soil profile. Soil profile is surface to bottom of rooting zone.
- gw_max (mm) Maximum available water holding capacity of ground water profile.
- inhibit_evap (flag) 0/1 enable/inhibit. Is indicator to when there is snowcover.

The following parameters are "declputparam" (i.e. the parameter values can be changed by this module.)

- rechr_ssr_K (m/d) daily ssr drainage from recharge.
- lower_ssr_K (m/d) daily ssr drainage from soil column.
- Sd_ssr_K (m/d) daily depression storage ssr drainage factor.
- Sd_gw_K (m/d) daily depression storage gw drainage.
- soil_gw_K (m/d) daily maximum amount of the soil water excess for an HRU that is routed directly to the associated groundwater reservoir each.
- gw_K (m/d) daily ground water drainage from gw reservoir.

Variable Inputs

- soil_rechr (*) (mm).
- soil_moist (*) (mm).
- gw (*) (mm).

SetSoil

Defined in ClassSetSoil. This module contains no active code. It performs the same functions as the original Classbasin but adds the capability of setting ClassSoil parameters from soil type and volumetric water content. It declares general parameters for the model. Examples of the parameters declared are: basin area, <u>HRU</u> area, latitude, elevation, ground slope (GSL) and aspect angle (ASL).

Observations

• none

Variables

• run_ID () - run identification.

Parameters

- basin_area (km^2) basin area.
- hru area (km^2) HRU area.
- hru_lat (°) latitude.
- hru_elev (m) altitude.
- hru_GSL (°) ground slope.
- hru_ASL (°) aspect.
- basin_name text string.
- hru_names text strings.
- RUN_ID integer number. Used to uniquely identify "CRHM_output" log files when RUN_ID is positive.
- RUN_START (d) run start day. Only used on batch execution.
- RUN_END (d) run end day. Only used on batch execution.
- INIT_STATE initial state file (Automation).
- Vol_h2o_content () fractional volumetric water content.
- soil_Depth_rechr (m) recharge depth.
- soil_Depth (m) soil column depth.
- soil_type_rechr () recharge layer soil type. 0 12, water/sand/loamsand/sandloam/loam/siltloam/sasclloam/clayloam/siclloam/

• soil_type () - soil column. 0 - 12, water/sand/loamsand/sandloam/loam/siltloam/sasclloam/clayloam/siclloam/siltloam/sasclloam/siltloam/sasclloam/siltloam/siltloam/sasclloam/siltloam/siltloam/sasclloam/siltloam/siltloam/sasclloam/siltloam/siltloam/sasclloam/siltloam/siltloam/sasclloam/siltloam/siltloam/siltloam/sasclloam/siltloam/siltloam/siltloam/siltloam/siltloam/sasclloam/siltloam/si

The following parameters are set by this module from the above information and need not be set.

- soil_rechr_max = soil_Depth_rechr*SetSoilproperties[soiltype][field capacity wilt].
- soil_rechr_init = soil_Depth_rechr*
 (Vol_h2o_content*SetSoilproperties[soiltype_rechr][field] SetSoilproperties[soiltype_rechr][wilt]).
- soil_moist_max = soil_Depth*SetSoilproperties[soiltype][field capacity wilt].
- soil_moist_init = soil_Depth*(Vol_h2o_content*SetSoilproperties[soiltype] [field] - SetSoilproperties[soiltype][wilt]).
- soil_rechr_init is limited to the range >= 0 and <= soil_rechr_max.
- soil_moist_init is limited to the range >= 0 and <= soil_moist_max.
- soil_rechr_init cannot be > soil_moist_init.
- When the two soil layers are of different types the values are only approximate.

Variable Inputs

• none

Notes

1. This module only executes once as an initialization routine before the model execution..

available (mm)	wilt (mm)	field capacity (mm)	pore space ()	
1000.0	0.0	1000.0	1000.0	0 water
84.0	40.0	124.0	395.0	1 sand
80.0	60.0	140.0	410.0	2 loamsand
130.0	100.0	230.0	435.0	3 sandloam

157.0	110.0	267.0	451.0	4 loam
162.0	130.0	292.0	485.0	5 siltloam
170.0	140.0	310.0	420.0	6 saclloam
167.0	150.0	317.0	476.0	7 clayloam
150.0	190.0	340.0	477.0	8 siclloam
150.0	200.0	350.0	426.0	9 sandclay
150.0	210.0	360.0	492.0	10 siltclay
145.0	215.0	360.0	482.0	11 clay

Volumetric

Defined in ClassVolumetric. The purpose of this module is to display the volumetric soil moisture as calculated from the variables in the module "Soil" and the properties of the soil. Additionally, on a selected date in the fall it sets the parameter " fallstat" handle the infiltration into frozen soil for the following spring as determined from the soil properties and the soil moisture variables in the module "Soil".

Observations

• none

Variables

• Volumetric () - fractional volumetric soil moisture and equals (soil_moist[hh]/soil_Depth[hh] + SetSoilproperties[soiltype[hh]] [1])/1000.0).

Parameters

- soil_Depth (m) soil column depth.
- soil_type () soil column. 0 12, water/sand/loamsand/sandloam/loam/siltloam/sasclloam/clayloam/siclloam/si
- set_fallstat () Julian day when the fallstat parameter is set from the current soil moisture

The following parameters are set by this module from the above information and need not be set.

• fallstat () - percentage of pore space holding water and equals Volumetric*100.0.

Variable Inputs

• soil_moist (*) (mm) - from module Soil etc.

Notes

1. During the first interval of execution, if set_fallstat is greater than the current Julian date, fallstat is set. Otherwise, only on the last interval of the actual Julian date.

available (mm)	wilt (mm)	field capacity (mm)	pore space ()	
1000.0	0.0	1000.0	1000.0	0 water
84.0	40.0	124.0	395.0	1 sand
80.0	60.0	140.0	410.0	2 loamsand
130.0	100.0	230.0	435.0	3 sandloam
157.0	110.0	267.0	451.0	4 loam
162.0	130.0	292.0	485.0	5 siltloam
170.0	140.0	310.0	420.0	6 saclloam
167.0	150.0	317.0	476.0	7 clayloam
150.0	190.0	340.0	477.0	8 siclloam
150.0	200.0	350.0	426.0	9 sandclay
150.0	210.0	360.0	492.0	10 siltclay
145.0	215.0	360.0	482.0	11 clay

frostdepth

This module is defined in Classfrostdepth. The number of soil layers is defined by NLAY. The snow cover is handled as a separate layer whose depth is calculated from the SWE of the HRU using the function <u>depthofsnow</u>.

Observations

• none

Variables

- frostdepth (m) frostdepth. State variable.
- snowdepth (mm) calculated from SWE. State variable.
- Findex (°C) cumulative degrees of frost in (day.°C). State variable.
- Tfreeze (d) days of frost. State variable.
- Lacc (MJ/m3) cumulative effective latent heat. State variable.
- Cacc (MJ/(m3.K)) cumulative effective heat capacity. State variable.
- Kacc (W/(m K)) cumulative effective thermal conductivity. State variable.

Parameters

- Ta (°C) annual air temperature.
- soil_type () 1/sand 2/loam 3/clay 4/organic (only used to differentiate between mineral and organic layers).
- por (m3/m3) porosity.
- theta (m3/m3) degree of saturation.
- d (m) layer depths 1 ... NLAY.
- hru_lat (°) latitude. Adjusts empirical Julian constants for Northern/Southern Hemisphere.

Variable Inputs

- hru_tmean (obs) (°C)
- SWE (pbsm) (mm)

Other Constants

- Lsnow = 0.35*Ci MJ/(m3) latent heat of snow.
- csnow = 0.35*ki + 0.65*ka MJ/(m3.°C) heat capacity of snow.
- ko = 0.21 W/(m K) organic material
- km = 0.27 W/(m K) mineral
- ka = 0.025 W/(m K) air
- ki = 2.24 W/(m K) ice
- kw = 0.57 W/(m K) water
- Cm = 2.000 MJ/(m3.K) mineral
- Cw = 4.185 MJ/(m3.K) water
- Ca = 0.001 MJ/(m3.K) air
- Co = 0.110 MJ/(m3.K) organic
- Ci = 1.950 MJ/(m3.K) organic

Returns

FrostDepth = sqrt((24*Kacc*Findex)/(Lacc + Cacc(Ta + Findex/(2.0*Tfreeze))))
- SnowDepth

where for individual layers

Ln = theta*por*333.0

Kn = (1.0 - por) * km + theta*por*kw + (por - theta*por)*ka

Cn = (1.0 - por) * Cm + theta*por*Cw + (por - theta*por)*Ca

for organic material use ko and Co instead of km and Cm.

For snow assuming a snowpack density of (250kg/m3) use

Ln = 0 Ks = 0.25*ki + 0.75*ka

Cs = 0.25*Ci

For the combined layers use

Lacc = (Ls*Ds + L1*D1 + L2*D2 ... Ln*Dn)/Dsum Cacc = (Cs*Ds + C1*D1 + C2*D2 ... Cn*Dn)/Dsum Kacc = Dsum/(Ds/Ks + D1/K1 + D2/K2 ... Dn/Kn)

where Ds (m) is SWE/250.0 assuming a snowpack density of (250kg/m3) and D1, D2 .. Dn, 1 ...NLAY are the respective layer thicknesses. The last layer only to the depth frozen.

Reference

Van Wijk W. R., (1963) Physics of Plant Environment. North-Holland Publishing Company - Amsterdam, pp.166

Netroute

This module defined in ClassNetroute, handles the routing of surface runoff, subsurface runoff and HRU routing using the lag and route method described by Clark(1945). Outflow from a HRU is calculated by lagging its inflow by the travel time through the HRU, then routing it through an amount of linear storage defined by the storage constant, K. The outflow from a HRU can be diverted to the inflow of another HRU or directed to the basin outlet. The parameter 'order' ensures that the outflows from the various HRUs are calculated in the correct order from upstream to downstream. The parameter 'whereto' defines the destination of an HRU outflow. A non zero value indicates flow to another HRU and a zero value specifies to the basin outflow.

The sum of runoff and snowmelt runoff may be delayed using independent Clark routing objects before being handled as input to the main HRU Clark object.

Similarly the subsurface runoff may also be delayed using an independent Clark object before being handled as input to the main HRU Clark routing object.

The delayed runoff and subsurface runoff are added and applied to the common HRU Clark routing object.

The output flow of an HRU may be diverted to another HRU by using the *whereto* parameter. This flow is used to top up the depression storage (Sd in the Soil module) and the soil moisture recharge layer (soil_rechr in the Soil module) of the HRU during the current time step. Any surplus is returned to the soil module by the *put* variable redirected_residual and is added to the soil_runoff variable during the next time step. The parameters soil_rechr_ByPass and Sd_ByPass may be used to prevent replenishing the recharge and depressional storage.

The groundwater flow from an HRU may be diverted to another HRU by using the 'gwwhereto' parameter. The gw flow is added to the input flow of the particular HRU after the recharge layer (soil_rechr in the Soil module) and the HRU free depressional storage (Sd in the Soil module) has been replenished. The parameters soil_rechr_ByPass and Sd_ByPass may be used to prevent replenishing the recharge and depressional storage.

Observations

• none

Variables

- inflow (mm*km²/int) sum of the optional inflow from other HRUs and delayed surface and subsurface runoff from this HRU.
- cuminflow (mm*km^2) HRU cumulative inflow.
- outflow (mm*km^2/int) HRU outflow after Clark(Kstorage, Lag) delay.
- cumoutflow (mm*km^2) HRU cumulative outflow.
- outflow_diverted (mm*km^2) HRU outflow diverted to another HRU.
- cumoutflow_diverted (mm*km^2) HRU cumulative outflow diverted to another HRU.
- cum_to_Sd (mm) cumulative flow from other HRUs to the depression storage of this HRU.
- cum_to_soil_rechr (mm) cumulative other HRU to soil_rechr of this HRU.
- ssrinflow (mm*km²/int) subsurface runoff (ssr) to this HRU.
- ssrcuminflow (mm*km^2) cumulative ssrinflow.
- ssroutflow (mm*km^2/int) ssr outflow after Clark(ssrKstorage, ssrLag) delay.
- ssrcumoutflow (mm*km^2) cumulative ssr outflow.
- runinflow (mm*km^2/int) sum of snowmelt runoff and runoff to this HRU.
- runcuminflow (mm*km²) cumulative runinflow.
- runoutflow (mm*km^2/int) runoff outflow after Clark(runKstorage, runLag) delay.
- runcumoutflow (mm*km^2) cumulative runoutflow.
- gwinflow (mm*km²) gw inflow.
- gwcuminflow (mm*km^2) cumulative gwinflow.
- gwoutflow (mm*km^2/int) gw outflow after Clark(gwKstorage, gwLag) delay.
- gwcumoutflow (mm*km^2) cumulative gw outflow.
- gwoutflow_diverted (mm*km^2/int) gw outflow diverted to other HRU.
- gwcumoutflow_diverted (mm*km^2/int) cumulative gw outflow diverted to other HRU.
- basinflow (m^3/int) average outflow rate of surface and subsurface runoff

from the basin.

- basinflow_s (m^3/s) average outflow rate of surface and subsurface runoff from the basin.
- cumbasinflow (m³) cumulative basinflow.
- basingw (m³/int) average outlow rate of groundwater from the basin.
- basingw_s (m^3/s) average outflow rate of groundwater from the basin.
- cumbasingw (m3) cumulative basingw.

Parameters

- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- Kstorage (d) storage constant for HRU.
- Lag (h) Lag for HRU.
- ssrKstorage (d) storage constant for subsurface runoff to HRU routing.
- ssrLag (h) Lag for subsurface runoff to HRU routing.
- runKstorage (d) storage constant runoff to HRU routing.
- runLag (h) Lag for runoff to HRU routing.
- whereto () '0' for basin outflow or 1..n to be directed toanother HRU.
- gwwhereto () 0 basin gw (basingw), >0 other HRU surface input. <0 other -HRU gw input, or (< -HRUmax or > +HRUmax) - surface basin outflow (basinflow),
- order () HRU routing process order.
- Sdmax (mm) maximum depression storage.
- soil_rechar_max (mm) soil recharge maximum.
- Sd_ByPass () 0 normal, 1 Bypass Pond/Depressional storage (i.e. Sd). Also applies to redirected gw.
- soil_rechr_ByPass () 0 normal, 1 Bypass recharge layer (i.e. soil_rechr). Also applies to redirected gw.
- gwKstorage (d) storage constant for gw to HRU routing.
- gwLag (h) Lag for gw to HRU routing.

Variable Inputs

• soil_ssr (*) (mm/int) - from module Soil etc. Programmed as an interval or

daily value.

- soil_runoff (*) (mm/int) from module Soil etc. Programmed as an interval or daily value.
- soil_gw (*) (mm/int) from module Soil etc. Programmed as a daily value.
- Sd (*) (mm) from module Soil etc. Programmed as a state *put* value.
- soil_rechr (*) (mm) from module Soil etc. Programmed as a state *put* value.
- soil_moist (*) (mm) from module Soil etc. Programmed as a state *put* value.
- redirected_residual (*) (mm*km^2/int) to module Soil etc. Programmed as a state *put* value.

Notes.

The redistribution of water between HRU's is handled by Netroute. The interval outflow from an HRU may be directed to another HRU or designated as basin flow using the whereto parameter. When directed to another HRU, any water is first used to top up the soil recharge layer if soil_rechr_ByPass is not enabled and then any surplus applied to Depression or Pond storage (Sd) if Sd_ByPass is not enabled and then the excess is added to the HRU's inflow. If the whereto parameter is 0, the outflow is accumulated in basin flow. The last HRU is basin flow, so whereto is always 0.

Netroute_D

This module defined in ClassNetroute_D, handles the routing of surface runoff, subsurface runoff and HRU routing using the lag and route method described by <u>Clark(1945)</u>. Outflow from a HRU is calculated by lagging its inflow by the travel time through the HRU, then routing it through an amount of linear storage defined by the storage constant, K. The outflow from a HRU can be diverted to the inflow of other HRUs and to the basin outlet. The parameter 'distrib_Route' determines the outflow of an HRU to the other HRUs. The parameter 'distrib_Basin' determines the amount of outflow of an HRU to the the basin outflow.

The sum of runoff and snowmelt runoff may be delayed using independent Clark routing objects before being handled as input to the main HRU Clark object.

Similarly the subsurface runoff may also be delayed using an independent Clark object before being handled as input to the main HRU Clark routing object.

The delayed runoff and subsurface runoff are added and applied to the common HRU Clark routing object.

The output flow of an HRU may be diverted to another HRU by using the 'distrib_Route' parameter. This flow is used to top up the depression storage (Sd in the Soil module) and the soil moisture recharge layer (soil_rechr in the Soil module) of the HRUs during the current time step. Any surplus is returned to the soil module by the *put* variable redirected_residual and is added to the soil_runoff variable during the next time step. The parameters soil_rechr_ByPass and Sd_ByPass may be used to prevent replenishing the recharge and depressional storage.

The groundwater flow from an HRU may be diverted to another HRU by using the 'gwwhereto' parameter. The gw flow is added to the input flow of the particular HRU after the recharge layer (soil_rechr in the Soil module) and the HRU free depressional storage (Sd in the Soil module) has been replenished. The parameters soil_rechr_ByPass and Sd_ByPass may be used to prevent replenishing the recharge and depressional storage.

Observations

• none

Variables

- inflow (mm*km²/int) sum of the optional inflow from other HRUs and delayed surface and subsurface runoff from this HRU.
- cuminflow (mm*km^2) HRU cumulative inflow.
- outflow (mm*km^2/int) HRU outflow after Clark(Kstorage, Lag) delay.
- cumoutflow (mm*km^2) HRU cumulative outflow.
- cum_to_Sd (mm) cumulative flow from other HRUs to the depression storage of this HRU.
- cum_to_rechr (mm) cumulative flow from other HRUs to the soil_rechar of this HRU.
- ssrinflow (mm*km²/int) subsurface runoff (ssr) to this HRU.
- ssrcuminflow (mm*km^2) cumulative ssrinflow.
- ssroutflow (mm*km^2/int) ssr outflow after Clark(ssrKstorage, ssrLag) delay.
- ssrcumoutflow (mm*km^2) cumulative ssr outflow.
- runinflow (mm*km^2/int) sum of snowmelt runoff and runoff to this HRU.
- runcuminflow (mm*km²) cumulative runinflow.
- runoutflow (mm*km^2/int) runoff outflow after Clark(runKstorage, runLag) delay.
- runcumoutflow (mm*km^2) cumulative runoutflow.
- gwinflow (mm*km 2) gw inflow.
- gwcuminflow (mm*km^2) cumulative gwinflow.
- gwoutflow (mm*km^2/int) gw outflow after Clark(gwKstorage, gwLag) delay.
- gwcumoutflow (mm*km^2) cumulative gw outflow.
- gwoutflow_diverted (mm*km^2/int) gw outflow diverted to other HRU.
- gwcumoutflow_diverted (mm*km^2/int) cumulative gw outflow diverted to other HRU.
- cumscaling_boost (mm*km^2) cumulative amout inflow boosted.
- basinflow (m^3/int) average outflow rate of surface and subsurface runoff from the basin.
- basinflow_s (m^3/s) average outflow rate of surface and subsurface runoff from the basin.
- cumbasinflow (m³) cumulative basinflow.
- basingw (m^3/int) average outlow rate of groundwater from the basin.

- basingw_s (m3/s) average outflow rate of groundwater from the basin.
- cumbasingw (m3) cumulative basingw.

Parameters

- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- Kstorage (d) storage constant for HRU.
- Lag (h) Lag for HRU.
- ssrKstorage (d) storage constant for subsurface runoff to HRU routing.
- ssrLag (h) Lag for subsurface runoff to HRU routing.
- runKstorage (d) storage constant runoff to HRU routing.
- runLag (h) Lag for runoff to HRU routing.
- distrib_Route () route HRU flow array. None zero values indicate diversion from this HRU to another HRU.
- distrib_Basin () HRU basin flow array. None zero values indicate diversion from this HRU to the basin flow.
- gwwhereto () 0 basin gw (basingw), >0 other HRU input. <0 other -HRU gw input, or (< -HRUmax or > +HRUmax) basin outflow (basinflow),
- Sdmax (mm) maximum depression storage.
- soil_rechr_max (mm) maximum depression storage.
- Sd_ByPass () 0 normal, 1 Bypass Pond/Depressional storage (i.e. Sd).
- soil_rechr_ByPass () 0 normal, 1 Bypass recharge layer (i.e. soil_rechr).
- scaling_factor () multiplies the inflow to Muskingum by this scaling factor. Experimental.

Variable Inputs

- soil_ssr (*) (mm/int) from module Soil etc. Programmed as an interval or daily value.
- soil_runoff (*) (mm/int) from module Soil etc. Programmed as an interval or daily value.
- soil_gw_D (*) (mm/d) from module Soil etc. Programmed as a daily value.

- Sd (*) (mm) from module Soil etc. Programmed as a state *put* value.
- soil_rechr (*) (mm) from module Soil etc. Programmed as a state *put* value.
- soil_moist (*) (mm) from module Soil etc. Programmed as a state *put* value.
- redirected_residual (*) (mm*km^2/int) to module Soil etc. Programmed as a state *put* value.

Notes.

The distrib_Route parameter if positive is used as is. However, if negative it is multiplied by the HRU area and made positive before being used. The last drainage HRU can only flow to the basin outlet. For the last drainage HRU the distrib_Route values must be zero..

The parameter distrib_Basin must always be zero or greater. If the last drainage HRU has distrib_Basin set to zero it is forced to one.

The redistribution of water between HRU's is handled by Netroute_D. The interval outflow from an HRU may be directed to another HRU or designated as basin flow using the distrib_Basin and distrib_Route parameters. When directed to another HRU, any flow is first used to top up the soil recharge layer (soil_rechr), if soil_rechr_ByPass is not set, then to top the Depression or Pond storage (Sd) if Sd_ByPass is not set and finally any excess is directed to the designated HRU's inflows. The absolute value of the parameter values determines the contribution to each destination. E.g, if the outflow of HRU 1 is Outflow1, then the diversion to HRU 2 would be N2*Outflow1/(B1 + N1 + N2 + N3 + N4 ...Nn), where B1 is the amount of HRU 1's output to basin flow and N1 to Nn the contribution of HRU1 to the other HRUs. Normally. Sinnce an HRU cannot recirculate its output to itself, in this example N1 would be zero.

Netroute_M

This module defined in ClassNetroute_M, handles the routing of surface runoff, subsurface runoff and HRU routing using Muskingum method. Outflow from a HRU is calculated by lagging its inflow by the travel time through the HRU, then routing it through an amount of linear storage defined by the storage constant, K. The outflow from a HRU can be diverted to the inflow of another HRU or directed to the basin outlet. The parameter 'order' ensures that the outflows from the various HRUs are calculated in the correct order from upstream to downstream. The parameter 'whereto' defines the destination of an HRU outflow. A non zero value indicates flow to another HRU and a zero value specifies to the basin outflow.

The sum of runoff and snowmelt runoff may be delayed using independent Clark routing objects before being handled as input to the main HRU Clark object.

Similarly the subsurface runoff may also be delayed using an independent Clark object before being handled as input to the main HRU Clark routing object.

The delayed runoff and subsurface runoff are added and applied to the common HRU Clark routing object.

The output flow of an HRU may be diverted to another HRU by using the *whereto* parameter. This flow is used to top up the depression storage (Sd in the Soil module) and the soil moisture recharge layer (soil_rechr in the Soil module) of the HRU during the current time step. Any surplus is returned to the soil module by the *put* variable redirected_residual and is added to the soil_runoff variable during the next time step. The parameters soil_rechr_ByPass and Sd_ByPass may be used to prevent replenishing the recharge and depressional storage.

The groundwater flow from an HRU may be diverted to another HRU by using the 'gwwhereto' parameter. The gw flow is added to the input flow of the particular HRU after the recharge layer (soil_rechr in the Soil module) and the HRU free depressional storage (Sd in the Soil module) has been replenished. The parameters soil_rechr_ByPass and Sd_ByPass may be used to prevent replenishing the recharge and depressional storage.

Observations

• none

Variables

- inflow (mm*km²/int) sum of the optional inflow from other HRUs and delayed surface and subsurface runoff from this HRU.
- cuminflow (mm*km^2) HRU cumulative inflow.
- outflow (mm*km^2/int) HRU outflow after Clark(Kstorage, Lag) delay.
- cumoutflow (mm*km^2) HRU cumulative outflow.
- outflow_diverted (mm*km^2) HRU outflow diverted to another HRU.
- cumoutflow_diverted (mm*km^2) HRU cumulative outflow diverted to another HRU.
- cum_to_Sd (mm) cumulative flow from other HRUs to the depression storage of this HRU.
- cum_to_soil_rechr (mm) cumulative other HRU to soil_rechr of this HRU.
- ssrinflow (mm*km²/int) subsurface runoff (ssr) to this HRU.
- ssrcuminflow (mm*km^2) cumulative ssrinflow.
- ssroutflow (mm*km^2/int) ssr outflow after Clark(ssrKstorage, ssrLag) delay.
- ssrcumoutflow (mm*km^2) cumulative ssr outflow.
- runinflow (mm*km^2/int) sum of snowmelt runoff and runoff to this HRU.
- runcuminflow (mm*km²) cumulative runinflow.
- runoutflow (mm*km^2/int) runoff outflow after Clark(runKstorage, runLag) delay.
- runcumoutflow (mm*km^2) cumulative runoutflow.
- gwinflow (mm*km²) gw inflow.
- gwcuminflow (mm*km^2) cumulative gwinflow.
- gwoutflow (mm*km^2/int) gw outflow after Clark(gwKstorage, gwLag) delay.
- gwcumoutflow (mm*km^2) cumulative gw outflow.
- gwoutflow_diverted (mm*km^2/int) gw outflow diverted to other HRU.
- gwcumoutflow_diverted (mm*km^2/int) cumulative gw outflow diverted to other HRU.
- basinflow (m^3/int) average outflow rate of surface and subsurface runoff

from the basin.

- basinflow_s (m^3/s) average outflow rate of surface and subsurface runoff from the basin.
- cumbasinflow (m³) cumulative basinflow.
- basingw (m³/int) average outlow rate of groundwater from the basin.
- basingw_s (m^3/s) average outflow rate of groundwater from the basin.
- cumbasingw (m3) cumulative basingw.

Parameters

- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- Lag (h) Lag for HRU.
- route_n () Manning roughness coefficient.
- route_R (() -hydraulic radius.
- route_S0 (() longitudinal channel slope.
- route_L (m) routing length.
- route_X_M () dimensionless weighting factor.
- ssrKstorage (d) storage constant for subsurface runoff to HRU routing.
- ssrLag (h) Lag for subsurface runoff to HRU routing.
- runKstorage (d) storage constant runoff to HRU routing.
- runLag (h) Lag for runoff to HRU routing.
- whereto () '0' for basin outflow or 1..n to be directed toanother HRU.
- gwwhereto () 0 basin gw (basingw), >0 other HRU input. <0 other -HRU gw input, or (< -HRUmax or > +HRUmax) basin outflow (basinflow),
- order () HRU routing process order.
- Sdmax (mm) maximum depression storage.
- soil_rechar_max (mm) soil recharge maximum.
- Sd_ByPass () 0 normal, 1 Bypass Pond/Depressional storage (i.e. Sd). Also applies to redirected gw.
- soil_rechr_ByPass () 0 normal, 1 Bypass recharge layer (i.e. soil_rechr). Also applies to redirected gw.
- gwKstorage (d) storage constant for gw to HRU routing.
- gwLag (h) Lag for gw to HRU routing.

Variable Inputs

- soil_ssr (*) (mm/int) from module Soil etc. Programmed as an interval or daily value.
- soil_runoff (*) (mm/int) from module Soil etc. Programmed as an interval or daily value.
- soil_gw (*) (mm/int) from module Soil etc. Programmed as a daily value.
- Sd (*) (mm) from module Soil etc. Programmed as a state *put* value.
- soil_rechr (*) (mm) from module Soil etc. Programmed as a state *put* value.
- soil_moist (*) (mm) from module Soil etc. Programmed as a state *put* value.
- redirected_residual (*) (mm*km^2/int) to module Soil etc. Programmed as a state *put* value.

Notes.

The redistribution of water between HRU's is handled by Netroute. The interval outflow from an HRU may be directed to another HRU or designated as basin flow using the whereto parameter. When directed to another HRU, any water is first used to top up the soil recharge layer if soil_rechr_ByPass is not enabled and then any surplus applied to Depression or Pond storage (Sd) if Sd_ByPass is not enabled and then the excess is added to the HRU's inflow. If the whereto parameter is 0, the outflow is accumulated in basin flow. The last HRU is basin flow, so whereto is always 0.

Netroute_M_D

This module defined in ClassNetroute_M_D, handles the routing of surface runoff, subsurface runoff and HRU routing using the Muskingum method. Outflow from a HRU is calculated by lagging its inflow by the travel time through the HRU, then routing it through an amount of linear storage defined by the storage constant, K. The outflow from a HRU can be diverted to the inflow of other HRUs and to the basin outlet. The parameter 'distrib_Route' determines the outflow of an HRU to the other HRUs. The parameter 'distrib_Basin' determines the amount of outflow of an HRU to the the basin outflow.

The sum of runoff and snowmelt runoff may be delayed using independent Clark routing objects before being handled as input to the main HRU Clark object.

Similarly the subsurface runoff may also be delayed using an independent Clark object before being handled as input to the main HRU Clark routing object.

The delayed runoff and subsurface runoff are added and applied to the common HRU Clark routing object.

The output flow of an HRU may be diverted to another HRU by using the 'distrib_Route' parameter. This flow is used to top up the depression storage (Sd in the Soil module) and the soil moisture recharge layer (soil_rechr in the Soil module) of the HRUs during the current time step. Any surplus is returned to the soil module by the *put* variable redirected_residual and is added to the soil_runoff variable during the next time step. The parameters soil_rechr_ByPass and Sd_ByPass may be used to prevent replenishing the recharge and depressional storage.

The groundwater flow from an HRU may be diverted to another HRU by using the 'gwwhereto' parameter. The gw flow is added to the input flow of the particular HRU after the recharge layer (soil_rechr in the Soil module) and the HRU free depressional storage (Sd in the Soil module) has been replenished. The parameters soil_rechr_ByPass and Sd_ByPass may be used to prevent replenishing the recharge and depressional storage.

Observations

• none

Variables

- inflow (mm*km²/int) sum of the optional inflow from other HRUs and delayed surface and subsurface runoff from this HRU.
- cuminflow (mm*km^2) HRU cumulative inflow.
- outflow (mm*km^2/int) HRU outflow after Clark(Kstorage, Lag) delay.
- cumoutflow (mm*km^2) HRU cumulative outflow.
- cum_to_Sd (mm) cumulative flow from other HRUs to the depression storage of this HRU.
- cum_to_rechr (mm) cumulative flow from other HRUs to the soil_rechar of this HRU.
- ssrinflow (mm*km²/int) subsurface runoff (ssr) to this HRU.
- ssrcuminflow (mm*km^2) cumulative ssrinflow.
- ssroutflow (mm*km^2/int) ssr outflow after Clark(ssrKstorage, ssrLag) delay.
- ssrcumoutflow (mm*km^2) cumulative ssr outflow.
- runinflow (mm*km^2/int) sum of snowmelt runoff and runoff to this HRU.
- runcuminflow (mm*km²) cumulative runinflow.
- runoutflow (mm*km^2/int) runoff outflow after Clark(runKstorage, runLag) delay.
- runcumoutflow (mm*km^2) cumulative runoutflow.
- gwinflow (mm*km 2) gw inflow.
- gwcuminflow (mm*km^2) cumulative gwinflow.
- gwoutflow (mm*km^2/int) gw outflow after Clark(gwKstorage, gwLag) delay.
- gwcumoutflow (mm*km^2) cumulative gw outflow.
- gwoutflow_diverted (mm*km^2/int) gw outflow diverted to other HRU.
- gwcumoutflow_diverted (mm*km^2/int) cumulative gw outflow diverted to other HRU.
- cumscaling_boost (mm*km^2) cumulative amout inflow boosted.
- basinflow (m^3/int) average outflow rate of surface and subsurface runoff from the basin.
- basinflow_s (m^3/s) average outflow rate of surface and subsurface runoff from the basin.
- cumbasinflow (m³) cumulative basinflow.
- basingw (m^3/int) average outlow rate of groundwater from the basin.

- basingw_s (m3/s) average outflow rate of groundwater from the basin.
- cumbasingw (m3) cumulative basingw.

Parameters

- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- Lag (h) Lag for HRU.
- route_n () Manning roughness coefficient.
- route_R (() -hydraulic radius.
- route_S0 (() longitudinal channel slope.
- route_L (m) routing length.
- route_X_M () dimensionless weighting factor.
- ssrKstorage (d) storage constant for subsurface runoff to HRU routing.
- ssrLag (h) Lag for subsurface runoff to HRU routing.
- runKstorage (d) storage constant runoff to HRU routing.
- runLag (h) Lag for runoff to HRU routing.
- distrib_Route () route HRU flow array. None zero values indicate diversion from this HRU to another HRU.
- distrib_Basin () HRU basin flow array. None zero values indicate diversion from this HRU to the basin flow.
- gwwhereto () 0 basin gw (basingw), >0 other HRU input. <0 other -HRU gw input, or (< -HRUmax or > +HRUmax) basin outflow (basinflow),
- Sdmax (mm) maximum depression storage.
- soil_rechr_max (mm) maximum depression storage.
- Sd_ByPass () 0 normal, 1 Bypass Pond/Depressional storage (i.e. Sd).
- soil_rechr_ByPass () 0 normal, 1 Bypass recharge layer (i.e. soil_rechr).
- scaling_factor () multiplies the inflow to Muskingum by this scaling factor. Experimental.

Variable Inputs

• soil_ssr (*) (mm/int) - from module Soil etc. Programmed as an interval or daily value.

- soil_runoff (*) (mm/int) from module Soil etc. Programmed as an interval or daily value.
- soil_gw_D (*) (mm/d) from module Soil etc. Programmed as a daily value.
- Sd (*) (mm) from module Soil etc. Programmed as a state *put* value.
- soil_rechr (*) (mm) from module Soil etc. Programmed as a state *put* value.
- soil_moist (*) (mm) from module Soil etc. Programmed as a state *put* value.
- redirected_residual (*) (mm*km^2/int) to module Soil etc. Programmed as a state *put* value.

Notes.

The distrib_Route parameter if positive is used as is. However, if negative it is multiplied by the HRU area and made positive before being used. The last drainage HRU can only flow to the basin outlet. For the last drainage HRU the distrib_Route values must be zero..

The parameter distrib_Basin must always be zero or greater. If the last drainage HRU has distrib_Basin set to zero it is forced to one.

The redistribution of water between HRU's is handled by Netroute_D. The interval outflow from an HRU may be directed to another HRU or designated as basin flow using the distrib_Basin and distrib_Route parameters. When directed to another HRU, any flow is first used to top up the soil recharge layer (soil_rechr), if soil_rechr_ByPass is not set, then to top the Depression or Pond storage (Sd) if Sd_ByPass is not set and finally any excess is directed to the designated HRU's inflows. The absolute value of the parameter values determines the contribution to each destination. E.g, if the outflow of HRU 1 is Outflow1, then the diversion to HRU 2 would be N2*Outflow1/(B1 + N1 + N2 + N3 + N4 ...Nn), where B1 is the amount of HRU 1's output to basin flow and N1 to Nn the contribution of HRU1 to the other HRUs. Normally. Sinnce an HRU cannot recirculate its output to itself, in this example N1 would be zero.

REW_route

This module defined in ClassREWroute, handles the routing of surface runoff, subsurface runoff from RBs (representative basins) and RB routing using the lag and route method described by Muskingum. Outflow from a RB is calculated by lagging its inflow by the travel time through the RB, then routing it through an amount of linear storage defined by the storage constant, K. The outflow from a RB can be diverted to the inflow of another RB or directed to the watershed outlet. The parameter 'order' ensures that the outflows from the various RBs are calculated in the correct order from upstream to downstream. The parameter 'whereto' defines the destination of an RB outflow. A non zero value indicates flow to another RB and a zero value specifies to the watershed outflow.

The delayed runoff and subsurface runoff are added and applied to the common RB routing object.

The flow output of an RB may be diverted to another RB by using the *whereto* parameter. Normally, this flow is added to the input flow of the particular RB and does not become part of the internal hydro;ogy of the RB.

This routine handles the groundwater from all RBs in an identical manner.

Observations

• none

<u>Variables</u>

- WS_inflow (m^3/int) sum of the optional inflow from other RBs and delayed surface and subsurface runoff from the current RB before Muskingum delay.
- cum_WSinflow (m³) HRU cumulative inflow.
- WS_outflow (m^3/int) HRU outflow after Muskingum delayed WS_inflow.
- cum_WSoutflow (m³) HRU cumulative outflow.
- WS_flow (m^3/int) average outflow rate of surface and subsurface runoff from the watershed.
- WS_flow_s (m^3/s) average outflow rate of surface and subsurface runoff

from the watershed.

- cum_WSflow (m^3) cumulative watershed outflow.
- WS_gwinflow (m³/int) sum of the optional inflow from other RBs and delayed surface and subsurface runoff from the current RB before Muskingum delay.
- cum_WSgwinflow (m^3) HRU cumulative inflow.
- WS_gwoutflow (m^3/int) HRU outflow after Muskingum delayed WS_gwinflow.
- cum_WSgwoutflow (m^3) HRU cumulative outflow.
- WS_gwflow (m^3/int) average outflow rate of surface and subsurface runoff from the watershed.
- WS_gwflow_s (m^3/s) average outflow rate of surface and subsurface runoff from the watershed.
- cum_WSgwflow (m^3) cumulative watershed outflow.

Parameters

- watershed_area (km²) Sum of RB areas.
- RB_area (km²) RB areas.
- route_n () Manning roughness coefficient.
- route_R () hydraulic radius.
- route_L (m) routing length.
- route_H (m) elevation change over route)L
- WS_Kstorage (d) storage constant for RB.
- WS_Lag (h) Lag for RB.
- WS_X_M () dimentionless weighting factor.
- WS_whereto () '0' for basin outflow or 1..n to be directed toanother HRU.
- WS_order () HRU routing process order.
- WS_gwKstorage (d) storage constant for RB.
- WS_gwLag (h) Lag for RB.
- WS_gwX_M () dimentionless weighting factor.
- WS_gwwhereto () '0' for basin outflow or 1..n to be directed toanother HRU.
- WS_gworder () HRU routing process order.

Variable Inputs

- inflow_All (*) (mm/int) "basinflow" variables from all groups. Programmed as an interval or daily value. Associated variable "rew".
- gw_All (*) (mm/int) "basingw" variables from all groups. Programmed as an interval or daily value. Associated variable "gwrew".

Notes.

The variable inputs above are generated by a special call to " cnt = declgrpvar(Variable name, "search variable", "help information", "units", var, var_data)". The name given as "search variable" is searched for in all model groups. The variable "cnt" gives the number of occurrences found. The one dimensional variable "var" gives the group index of the occurrences found. Group A - 1, B - 2, etc. The two dimensional [RBindex][HRU], variable"var_data" gives the interval data. In the case of "basinflow" and "basingw" which are dimensioned as "basin", [HRU] is always [0].

Muskingum Routing.

The following is calculated from each RB's parameter and used for the Muskingum routing.

Vw[3] = {1.67, 1.44, 1.33}; // rectangular - 0/parabolic - 1/triangular - 2

WS_Kstorage = route_L/(Vw[route_Cshp]*Vavg) and

WS_gwKstorage = route_L/(Vw[route_Cshp]*Vavg) where

Vavg = (1/route_n)*pow(route_R, 2/3)*pow(route_H/route_L, 0.5)*86400.

Demo Project.

watershed.prj

tsurface

A module to calculate skin surface temperature from hru_t and SWE. The basic module uses a 'Radiative-Conductive-Convective approach to surface temperature, T.J. (Williams et al, 2015). The variation#1 uses the parameter n_factor to estimate surface temperature from the air temperature. Woo (2012, p. 57).

Observations

• none

Variables

- hru_tsf (°C) surface/skin temperature.
- hru_tsfD (°C) mean daily surface/skin temperature.
- hru_tsf_acc (°C) temperature accumulator used to calculate the daily mean hru_tsfD.

Parameters

- W_a () empirical constant for air temperature. Only uses in tsurface.
- W_b () -empirical constant for net radiation at ground surface. Only uses in tsurface.
- W_c () empirical constant for frost table. Only uses in tsurface.
- W_d () empirical constant for frost table. Only uses in tsurface.
- n_factor_a () a constant in n_factor = n_factor_a n_factor_b*exp(n_factor_c*T) where T is days after start of ground thaw. Only uses in tsurface#1.
- n_factor_b () a constant in n_factor = n_factor_a n_factor_b*exp(n_factor_c*T) where T is days after start of ground thaw. Only uses in tsurface#1.
- n_factor_c () a constant in n_factor = n_factor_a n_factor_b*exp(n_factor_c*T) where T is days after start of ground thaw. Only uses in tsurface#1.
- n_factor_d () a constant in n_factor = n_factor_a n_factor_b*exp(n_factor_c*T) where T is days after start of ground thaw. Only uses in

tsurface#1

Variable Inputs

- hru_t (°C) air temperature
- SWE (mm) snow water equivalent.
- net (MJ/m²*int) net radiation. Only uses in tsurface.
- Zdt (m) active thawing front. Only uses in tsurface.

Notes

Qmelt

This module defined in Classqmelt to calculate the ground flux for hummuckcovered hillslopes in the Artic tundra.

Observations

- ta(°C) average air temperature during time step. In original version was t but was changed to prevent conflict with CRHM.
- ts(°C) average surface/skin temperature during time step.

Function Observations

- tmean (°C) mean daily air temperature.
- tpos (°C) sum of interval air temperatures above 0°C during a day.
- tsmean (°C) mean daily surface/skin temperature.
- tspos (°C) sum of interval surface/skin temperatures above 0°C during a day.

Variables

- Qm (MJ/m2 ∆t) snow melt during the time step, calculated using the mean daily air temperature.
- cum_Qm (MJ/m2 △t) cumulative snow melt (from daily mean air temperature) over time.
- Qg (MJ/m2) -ground heat flux during the time step, calculated using the mean daily surface/skin temperature.
- cum_Qg (MJ/m2) cumulative ground heat flux (from daily mean surface/skin temperature) over time.

Parameters

- TIa0Mn (°C) offset in temperature index model in calculating QmD.
- TIa1Mn (MJ/day/°C) coefficient for estimating QmD.
- TIs0Mn (°C) offset in temperature index model in calculating QgD.
- TIs1Mn (MJ/day/°C) coefficient for estimating QgD.

Variable Inputs

• none

Notes

Snow melt from mean daily air temperature .

- 1. if (tmean TIa0Mn > 0.0) QmD = (tmean TIa0Mn)*TIa1Mn,
- 2. if time step $t > 0.0 \text{ Qm} = t^{*}\text{QmD/tpos}$ else Qm = 0.0.
- 3. $cum_Qm = cum_Qm + Qm$.

Ground heat flux from mean daily surface/skin temperature

- 1. if (tsmean TIs0Mn > 0.0) QgD = (tsmean TIs0Mn)*TIs1Mn,
- 2. if time step ts > 0.0 Qg = ts*QgD/tspos else Qg = 0.0.
- 3. $\operatorname{cum}_Q g = \operatorname{cum}_Q g + Qg$.

Qmelt2

This module defined in Classqmelt to calculate the ground flux for hummuckcovered hillslopes in the Artic tundra. CRHM compatible.

Observations

• none

Variables

- Qm (MJ/m2 △t) snow melt during the time step, calculated using the mean daily air temperature.
- cum_Qm (MJ/m2 △t) cumulative snow melt (from daily mean air temperature) over time.
- Qg (MJ/m2) -ground heat flux during the time step, calculated using the mean daily surface/skin temperature.
- cum_Qg (MJ/m2) cumulative ground heat flux (from daily mean surface/skin temperature) over time.

Parameters

- TIa0Mn (°C) offset in temperature index model in calculating QmD.
- TIa1Mn (MJ/day/°C) coefficient for estimating QmD.
- TIs0Mn (°C) offset in temperature index model in calculating QgD.
- TIs1Mn (MJ/day/°C) coefficient for estimating QgD.

Variable Inputs

- hru_t (°C) air temperature.
- hru_tmean (°C) mean air temperature.
- hru_tsfD (°C) daily mean surface/skin temperature.

Notes

Snow melt from mean daily air temperature .
- 1. if (tmean TIa0Mn > 0.0) QmD = (tmean TIa0Mn)*TIa1Mn,
- 2. if time step $t > 0.0 \text{ Qm} = t^*\text{QmD/tpos}$ else Qm = 0.0.
- 3. $\operatorname{cum}_Qm = \operatorname{cum}_Qm + Qm$.

Ground heat flux from mean daily surface/skin temperature

- 1. if (tsmean TIs0Mn > 0.0) QgD = (tsmean TIs0Mn)*TIs1Mn,
- 2. if time step ts > 0.0 Qg = ts*QgD/tspos else Qg = 0.0.
- 3. $\operatorname{cum}_Q g = \operatorname{cum}_Q g + Qg$.

Qdrift

This module defined in Classqdrift calculates the amount of runoff that originates from melting of deep snow drifts on hummock-covered hillslopes in the Arctic tundra. CRHM compatible. Basic module uses the observation p. Variation #1 uses the variable hru_p from the module obs.

Observations

• p (mm/int) - precipitation. Assumed to be rainfall. Basic module.

Variables

- SWE (mm) mean water equivalent of snowcover.
- driftmelt (m3/int) volume of snowmelt released by the drift in the time step.
- driftmeltD (m3) daily snow melt from drift.
- cumdriftmelt (m3) cumulative snow melt from drift.

Parameters

- Type () HRU land type, 0=NOTUSED/1=DRIFT/2=HUMMOCK. Not used in Variation#1.
- length (m) length of HRU.
- DrainTo () the Hummock HRU drained to, 0=NOWHERE or HRU# (1 to MAXHRU).
- InitSWE (mm) initial mean snow water equivalent. Assumes drift has a triangular distribution in depth (x-section) oriented normal to creek. All exposed area occurs on the down slope edge.
- meltLag (hours) lag inserted into driftmelt.
- meltstorage (days) storage inserted into driftmelt.
- catchadjust () precipitation calibration.

Variable Inputs

- Qm (MJ/m2/int) snowmelt from module **Qmelt**.
- hru_p (mm/int) precipitation. Variation#1.

Notes

The snow is assumed to be distributed over the length of the HRU as a triangular distribution. The altitude of the triangle is twice the initial average snow water equivalent and the base the length of the HRU. As the snow melts, the triangle keeps the same proportions with all the melt occurring normal to the valley bottom, i.e. as the SWE depth decreases the snow covered base decreases in proportion.

To avoid round off errors the change in SWE is handled as follows: Note that the dimension of length (l) of the drift HRU is in the units.

```
float lastcumdriftmelt = cumdriftmelt[hh]; // following avoids
round off error
cumdriftmelt[hh] = length[hh]*InitSWE[hh]*(1.0 -
sqr(SWE[hh]/InitSWE[hh])); // (m3*1E3)
driftmelt[hh] = (cumdriftmelt[hh] - lastcumdriftmelt)/1E3; //
(m3/Interval)
```

Quinton

This module defined in Classquinton handles the subsurface drainage from hummock-covered hillslopes in the Arctic tundra. An HRU is a one-metre wide strip of land having defined hydrological properties that is aligned perpendicular to a selected channel. All measurements of depth are referenced using the ground surface as the datum unless otherwise stated and taken positive.

Observations

 p (mm △t) – depth of precipitation (rainfall) received at the ground surface in time step. In basic module only.

Variables

- theta (m3/m3) volumetric soil moisture of layer.
- layerwater (m) depth of liquid water in layer. It consists of melt, precipitation and inflow received from other HRUs.
- dmelt (m) depth to the frost table.
- wdrained (m) depth to the frozen saturated layer when dmelt is in the frozen unsaturated layer above. Otherwise 0.0.
- Depth (m) height of liquid water above the frost table.
- watertable (m) depth from the surface to the surface of the water table. Is equal to (dmelt Depth).
- d_surface (m) depth from the surface to the middle of the liquid water layer. This depth is used to determine the average horizontal hydraulic conductivity. Is equal to (dmelt Depth/2).
- k (m/day) HRU horizontal hydraulic conductivity.
- flow (m Dt) volume of subsurface flow per unit plan area of hummock HRU in a time step expressed as an average depth of

water (m3/m2)

- flowm3 (m3) volume of subsurface flow from hummock HRU in time step.
- cumflow (m) cumulative volume of subsurface flow per unit plan area of hummock HRU – expressed as an average depth of water (m3/m2).
- runoff (m Dt) volume of surface runoff per unit plan area of hummock HRU in a time step – expressed as an average depth of water (m3/m2).
- runoffm3 (m3/int) volume of surface runoff from hummock HRU in time step.
- cumrunoff(m/m2) cumulative volume of surface runoff per unit plan area of hummock HRU – expressed as an average depth of water (m3/m2).
- flowin (m/int) volume of inflow from external sources (melt, drainage, other) entering an unit plan area of a hummock HRU in a time step expressed as an average depth of water (m3/m2).
- flowinm3 (m3/int) volume of inflow from external sources (melt, drainage, other) entering a hummock HRU in a time step.
- cumflowin (m/m2) cumulative volume of inflow from external sources (melt, drainage, other) entering an unit plan area of a hummock HRU – expressed as an average depth of water (m3/m2).
- loss (m Dt) loss of water from all layers of hummock HRU per unit plan area in a time step– expressed as an average depth of water (m3/m2) over the time step.
- capillary (m) –depth of the capillary water in the layer.
- tension (m) head of water in the layer.
- transit (day) time for a particle of water to traverse an HRU. It is

equal to the length of the HRU.

- Cvis (J/m3/K) heat capacity of layer saturated, frozen condition.
- Cvisa (J/m3/K) heat capacity of layer in unsaturated, frozen condition.
- Cvws (J/m3/K) heat capacity of layer in saturated, unfrozen condition.
- Cvwsa (J/m3/K) heat capacity of layer in unsaturated, unfrozen condition.
- lamis (W/m/K) thermal conductivity of layer in saturated, frozen condition.
- lamws (W/m/K) thermal conductivity of layer in saturated, unfrozen condition.
- lamwsa (W/m/K) thermal conductivity of layer in saturated, unfrozen condition

Parameters

- Type () HRU land type, 0=NOTUSED, 1=DRIFT, 2=HUMMOCK.
- length (m) length of HRU. HRU has unit width.
- K_btm (m/d) horizontal bottom hydraulic conductivity.
- K_top (m/d) horizontal top hydraulic conductivity.
- ztrn (m) transition depth.
- DrainTo () If drift HRU, the Hummock HRU drained to, 0=NOWHERE or HRU# (1 to MAXHRU).
- soil_type() 0=organic1, 1=organic2, 2=organic3, 3=sand, 4=clay.
- tinit (°C) initial layer temperatures.
- slope (m/m) HRU average slope.
- Residual () organic non-drainable porosity. Also referred to as specific yield.
- d (m) soil layer thickness.
- Drained (m) depth of water table, i.e. saturated organic material (ice or water).

- FrozenTo (m) initial depth of frost table.
- flowLag (hours) lag inserted into flow out.
- flowstorage (days) storage inserted into flow out.
- runoffLag (hours) lag inserted into runoff.
- runoffstorage (days) storage inserted into runoff.
- Pors (m3/m3) used to redefine the porosity a soil type if the value is greater than zero. Affects all HRU's using this soil type.
- n () an empirical constant used in the Van Genuchten calculation. If zero Van Genuchten is not used to determine the soil tension.
- a () an empirical constant used in the Van Genuchten calculation.

Variable Inputs

- Qg (MJ/m2/ Δt) surface ground flux from module **Qmelt**.
- driftmelt $(m3/\Delta t)$ drift melt and precipitation from module **Qdrift**.
- hru_p (mm/int) precipitation in Variation#1

Heat capacity of layer soil/organic matter:

frozen/saturated

Cvis = Cv_i*por_s[soil_type_lay] + Cv_s[soil_type_lay]*(1.0-por_s[soil_type_lay]), (J/kg/K).

frozen/unsaturated - drained to the irreducible water content

Cvisa = Cv_i*por_s[soil_type_lay]*Residual + Cv_s[soil_type_lay]*(1.0por_s[soil_type_lay]) + Cv_a*(por_s[soil_type_lay] - Residual), (J/kg/K).

unfrozen/saturated

Cvws = Cv_w*por_s[soil_type_lay]) + Cv_s[soil_type_lay]*(1.0-por_s[soil_type_lay]), (J/kg/K).

unfrozen/unsaturated - drained to the irreducible water content

Cvwsa = Cv_w*Residual *por_s[soil_type_lay] + Cv_s[soil_type_lay]*(1.0-

por_s[soil_type_lay]) + Cv_a*(por_s[soil_type_lay] - Residual), (J/kg/K).

Thermal capacity of layer soil/organic matter:

Xs = 1.0 - por_s[soil_type_lay]

Xw = por_s[soil_type_lay]-Residual

Xa = 1.0 - Xs - Xw

n = por_s[soil_type_lay]

if(Xw >= 0.09)

ga = 0.333-Xa/n*(0.333-0.035)

else

ga = 0.013 + 0.944 * Xw

gc = 1.0 - 2.0*ga

$$\label{eq:Fs} \begin{split} Fs &= 1.0/3.0*(2.0/(1 + (ks_s[soil_type_lay]/lam_w-1.0)*0.125) + (1.0/((1 + (ks_s[soil_type_lay]/lam_w-1.0)*0.75)))) \end{split}$$

 $\label{eq:Fa} Fa = 1.0/3.0*(2.0/(1 + (lam_a/lam_w-1.0)*ga) + (1.0/((1 + (lam_a/lam_w-1.0)*gc))))$

a = Farouki_a(por_s[soil_type_lay]) sol fractional porosity = $3a^2 - 2a^3$

solution of the relationship:

frozen/saturated

lamis_lay = lam_i*a*a + lam_s[soil_type_lay]*sqr(1.0-a) +
lam_s[soil_type_lay]*lam_i*(2*a-2*sqr(a))/(lam_s[soil_type_lay]*a +
lam_i*(1.0-a))

unfrozen/saturated

lamws_lay = lam_w*a*a + lam_s[soil_type_lay]*sqr(1.0-a) + lam_s[soil_type_lay]*lam_w*(2*a-2*sqr(a))/(lam_s[soil_type_lay]*a+lam_w*(1.0-a))

unfrozen/unsaturated

lamwsa_lay = (Xw*lam_w + Fa*Xa*lam_a
+Fs*Xs*lam_s[soil_type_lay])/(Xw + Fa*Xa + Fs*Xs)

Calculation of melt

Latent heat of fusion, Hf = 334.4E+3 (J/kg).

frozen/saturated

melt = Qm/(-tlayer_lay *Cvis + por_s[soil_type_lay]*Hf)/1000 (m),

frozen/unsaturated - drained to the irreducible water content

melt = Qm/(-tlayer_lay *Cvisa + Residual*Hf)/1000 (m)

Assuming drainage is controlled by the Van Genuchten estimation.

It is assumed that when the soil tension is less than the bubbling pressure that the soil drains by gravity. Otherwise is controlled by:

```
\theta_f = (\phi - \theta_r)^* (2^n)^{-m} + \theta_r where \psi_b = 1/\alpha. Refer to <u>capillary</u> for additional information.
```

For the unfrozen depth of a soil layer and the liquid water in the thawed layer the actual soil moisture is calculated. If $\theta \ge \theta_f$, all of the melt is transferred to the variable, layer_water_lay otherwise, the amount transferred is determined by the soil tension and the Van Genuchten estimation.

```
excess = capillary_lay - θ<sub>f</sub> * d_lay /por_s
```

Assuming all melt drains by gravity

layerwater_lay = layerwater_lay + melt distributed over applicable layers.

Depth of free liquid water in organic layers;

Depth = Sum((layerwater_lay)) (m).

Depth from surface of the water table;

d_surface = dmelt - Depth/2.0 (m).

Calculation of Horizontal Hydraulic Conductivity.

Three power series were fitted to the horizontal hydraulic conductivity versus depth data collected at Siksik, Granger Basin and Scotty Creek sites. One series used all the data and the other two power series were created using only the data greater than the first power series trendline and the second series used only the values less than the first series trendline. These two power series were created to simulate the two extremes, i.e. minimum and maximum possible horizontal hydraulic conductivities.

The continuous function for hydraulic conductivity k are:

```
\log(K(z)) = \log(Kbtm) - (\log(Ktop) - \log(Kbtm))/(1 + (z/ztrn)^n)
```

Calculation of Horizontal Flow.

lossD = k*Depth*slope (m/day/m2).

loss = k*Depth*fall/length/Freq (m/m2/int). Where FREQ is the number of intervals/day.

Water balance.

The horizontal drainage is taken from the uppermost layerwater first and any left is taken from the next lower layer.

layerwater_lay = layerwater_lay - Loss/(por_s[soil_type_lay]*(1-Residual))

(m)

The out flow is calculated from the losses in the layerwater layers;

flow = Sum(Loss) (m).

N.B. units of (m/m2) are used for convenience.

Replenishment from snow drift melt;

```
flowin = driftmelt*length_drift_hru/length_hummock_hru (m) or from a hummock HRU is
```

flowin = flow*length_contributing_hummock_hru/length_hummock_hru.

layer maximum replenish depth:

maxdepth = dmelt - top_of_layer,

maximum replenish water (m),

maxfree = maxdepth*por[soil_type_lay]*(1.0 - replenish),

surplus after satisfying maxfree for melted layers is runoff (m).

Calculation of theta for each layer.

```
layerwater_ht = sum(layerwater_lay/por[soil_type_lay] (m),
```

if(dmelt - layerwater_ht) <= top_of_layer then</pre>

theta = 1.0

else if(dmelt - layerwater_ht) > (top_of_layer + layer_depth) then

theta = residual

else

theta = (depth_w +(layer_depth - depth_w)*residual)/layer_depth,

where depth_w = top_of_layer + layer_depth - (dmelt free_water_ht/por[soil_type_lay] (m).

Constants

g = 9.81 (m/s2)

visc_w = 0.0018 at 0°C (N.s/m2) (kg/(m.s) water

rho_a = 1.2 (kg/m3) air

rho_i = 920.0 (kg/m3) ice

rho_w = 1000.0 (kg/m3) water

c_a = 1010.0 (J/kg/K) air

c_i = 2120.0 (J/kg/K) ice

c_w = 4185.0 (J/kg/K) water

 $Cv_a = 1212.0$ (J/m3/K) air

Cv_i = 1950400.0 (J/m3/K) ice

Cv_w = 4185000.0 (J/m3/K) water

 $lam_a = 0.025 (W/m/K) air$

lam_i = 2.24 (W/m/K) ice

 $lam_w = 0.57$ (W/m/K) water

order[LOAM1, LOAM2, LOAM3, SAND, CLAY]

rho_s[] = { 41.1, 75.2, 91.4, 1300.0, 1300.0} (kg/m3) density

c_s[] = {1920.0, 1920.0, 1920.0, 890.0, 890.0} (J/m3/K) specific heat

Cv_s[] = {78912.0, 144384.0, 175392.0, 1157000.0, 1157000.0} (J/m3/K) heat capacity

lam_s[] = {0.21, 0.21, 0.21, 2.50, 2.50} (W/m/K) thermal conductivity ks_s[] = {450.0, 154.0, 13.0, 5.0, 3.0} (m/day) hydraulic conductivity por_s[] = {0.96, 0.9, 0.87, 0.43, 0.43, } () porosity

Notes on HRUs.

The Quinton modules are not areal models but written to handle a one-metre wide strip of land that is aligned perpendicular to a selected channel The strip is made up of , each having a distinct personality that is determined by the parameters assigned to it. At present there are two types of HRUs', **drift** and **hummock**. Any number of **drift** and **hummock** HRUs can be included in a model. **Drift** HRUs must drain into **hummock** HRUs, not into another **drift** HRU. Multiple **drift** HRUs can drain into one **hummock** HRU. **Drift** HRUs must be defined before the **hummock** HRUs they drain into. **Hummock** HRUs may be cascaded, draining into one another before finally draining into the creek. At present it is assumed that if an areal water balance is required for a basin, the ouput of the CRHM model will be further processed in a spreadsheet.

Water Balance.

To confirm proper operation of the modules a water balance was added to the Quinton and Qdrift modules. When the model is initialised at the beginning of a model run the water content defined in the initial conditions is tabulated in the Log/Debug Output window. Similarly at the completion of the model run the final water content is tabulated. An example of the results of a model run follow.

The outputs from the HRUs are given twice. Once, as the volume (m3) from the HRU given that the area is equal to the length of the HRU times one metre width and secondly as the equivalent water depth over the area of the HRU.

02 05 19 00 30 Initial

HRU 1: (Drift) - water content (m3) (m/m2): 5.40 0.200

HRU 2: (Hummock) - water content (m3) (m/m2): 4.16 0.297

HRU 3: (Hummock) - water content (m3) (m/m2): 4.16 0.297

HRU 4: (Hummock) - water content (m3) (m/m2): 4.16 0.297

HRU 5: (Hummock) - water content (m3) (m/m2): 4.16 0.297

HRU 6: (Hummock) - water content (m3) (m/m2): 4.16 0.297

HRU 7: (Hummock) - water content (m3) (m/m2): 4.16 0.297

HRU 8: (Hummock) - water content (m3) (m/m2): 4.16 0.297

02 08 01 00 00 Final

HRU 1: (Drift) - water content (m3) (m/m2): 0.00 0.000

HRU 1: (Drift) - total precip (m3) (mm/m2): 0.00 0.000

HRU 1: (Drift) - water storage (m3) (m/m2): 0.00 0.000

HRU 2: (Hummock) - water content (m3) (m/m2): 3.11 0.222

HRU 2: (Hummock) - cumulative flowin (m3) (m/m2): 5.40 0.386

HRU 2: (Hummock) - cumulative precip (m3) (m/m2): 0.00 0.000

HRU 2: (Hummock) - cumulative flowout (m3) (m/m2): 3.90 0.278

HRU 2: (Hummock) - flowout in storage (m3) (m/m2): 0.00 0.000

HRU 2: (Hummock) - cumulative runoff (m3) (m/m2): 2.55 0.182

HRU 2: (Hummock) - runoff in storage (m3) (m/m2): 0.00 0.000

HRU 3: (Hummock) - water content (m3) (m/m2): 3.11 0.222

HRU 3: (Hummock) - cumulative flowin (m3) (m/m2): 6.45 0.460

HRU 3: (Hummock) - cumulative precip (m3) (m/m2): 0.00 0.000

HRU 3: (Hummock) - cumulative flowout (m3) (m/m2): 5.43 0.388 HRU 3: (Hummock) - flowout in storage (m3) (m/m2): 0.00 0.000 HRU 3: (Hummock) - cumulative runoff (m3) (m/m2): 2.06 0.147 HRU 3: (Hummock) - runoff in storage (m3) (m/m2): 0.00 0.000

Processing of parameters Drained and FrozenTo.

When the frost table is below the water table the initialization consists of moving the water contained between the two tables into the variable layerwater_lay??? for the applicable layers. When the (frozen) water table is below the frost table the quantity of heat required to melt the frozen soil is less as instead of saturated s. **Don't understand?? Do you mean?** When the water table falls below the frost table the quantity of heat required to melt the drained soil is less because its' moisture content is set to the residual value

	Fibric	Hemic	Sapric
k _s (m/s) - hydraulic conductivity.	1.7 *	2.0 *	1.0 *
	10 ⁻⁴	10 ⁻⁶	10 ⁻⁷
$\theta_{\rm P} \ ({\rm m}^{3}/{\rm m}^{3})$ - porosity.	0.93	0.88	0.83
θ_{LIM} (m ³ /m ³) - residual water content.	0.04	0.15	0.22
S _y (m ³ /m ³) - specific yield.	0.5	0.3	
$\psi_{S}(cm)$ - suction at saturation.	1.03	1.02	1.01
h () - soil texture exponent.	2.7	6.1	12

Derived from Letts et al. (2000).

glacier

This module is defined in Classglacier and is an energy-budget firn and ice melt model loosely based on work done by (Gray and Landine, 1987) for the Canadian Prairies. However, it has variations allowing it to input the melt input energy from other CRHM modules. The SWE is handled externally by the module "SnobalCRHM".

Use with module "pbsmCRHM". Normally uses internal melt model based upon module "ebsm" but by using variations 1 use "Qnsn_Var (variable)", variation 2 use "Qnsn (observation)" and with basic and variation 3 use "QnD (observation)".

Observations

- QnD (MJ/m² day) net radiation. Basic module glacier and glacier#3. Always optional.
- Qnsn (W/m^2 day) net all-wave at snow surface. Variation #2.

Variables

- firn_h (mm) depths of snow densification layers.
- firn_dens (1000 kg/m^3) densities of snow densification layers.
- ice_init (mm) Initial water equivalent of glacier ice calculated from firn_h and firn_den.
- firn_yr (yr) years to 'critical density' and age of firn.
- ice (mm) Water equivalent of ice. State variable.
- firn (mm) Water equivalent of all firn layers. State variable.
- glacier_all (mm) water equivalent of SWE, firn layers and ice.
- firn_depth (mm) depth of all firn layers.
- firn_init (mm) initial water equivalent of all firn layers.
- nfirn () number of layers used in firn densification.
- snow_ice_firnmeltD (mm/d) daily snowmelt applied to SWE, firn and ice.
- cumsnow_ice_firnmelt (mm) cummulative snowmelt applied to SWE, firn and ice.
- GlacierMode () 1 glacier, 0 tongue (no firn or ice).
- SWEmelt (mm/d) daily snowmelt.

- SWEmelt_delayed (mm/d) daily snowmelt delayed by Clark.
- cumSWEmelt (mm) cummulative firn snowmelt.
- firnmelt (mm/d) daily firn melt.
- firnmelt_delayed (mm/d) daily firn melt delayed by Clark.
- cumfirnmelt (mm) cummulative firn melt.
- icemelt (mm/d) daily ice melt.
- icemelt_delayed (mm/d) daily ice melt delayed by Clark.
- cumicemelt (mm) cummulative ice melt.
- net_rain_org (mm/int) contents of net_rain saved before the addition of SWE, firn and ice melt to original contents.
- LW_ebsm (mm/d) liquid water in snowpack. State variable.
- u_ebsm (MJ/m^2*d) snowpack energy deficit. State variable.
- Gmelt (MJ/m^2*d) Qmelt = Qn+Qh+Qe+Qp.
- Gn_ebsm (MJ/m^2*d) net radiation. Changed from Gn_ebsm to avoid name conflict.
- Gh_ebsm (MJ/m^2*d) sensible heat. Changed from Qn_ebsm to avoid name conflict.
- Ge_ebsm (MJ/m^2*d) latent heat. Changed from Qn_ebsm to avoid name conflict.
- Gp_ebsm (MJ/m^2*d) input from rainfall. Changed from Qn_ebsm to avoid name conflict.
- SWE_2firn (mm) end of summer SWE transferred to firn.
- SWE_2firn_dens (kg/m^3) end of summer density of SWE transferred to firn
- cumSWE_2firn (mm) cummulative SWE added to firn..
- LWmax (mm) -maximum liquid water in snowpack.
- net_rainD (mm/d) daily net rain calculated from net_rain.
- Qnsn_Acc (W/m^2) accumulator for Qnsn_Var/Qnsn. Variation glacier#1 and #2
- topo_elev (m) topography elevations below glacier.
- hru_elev_value (m) topography elevations below glacier.
- snowmeltD_to_net_rain (m/d) snowmelt/firn/ice directed to be handled like rain.
- cumsnowmeltD_to_net_rain (m) cummulative snowmelt/firn/ice directed to be handled like rain.

Parameters

• delay_melt () - inhibits any melt before this date. Only used when tfactor

and nfactor are both zero. Default value not suitable for Southern Hemisphere

- basin_area (km^2) basin area.
- hru area (km²) HRU area.
- Elev_Adj_glacier_surf () enable add depth of glacier to land elevation (hru_elev).
- ice_init (mm) Initial water equivalent of glacier ice.
- ice_dens (kg/m^3) density of glacier ice.
- firn_h_init (mm) initial depths of firn densification layers.
- firn_dens_init (1000 kg/m^3) initial densities of firn densification layers.
- firn_Albedo () Initial glacier firn albedo.
- ice_Albedo () Initial glacier ice albedo.
- SWEstorage (d) Storage constant for SWE.
- SWElag (h) Lag constant for SWE.
- firnstorage (d) Storage constant for firn.
- firnlag (h) Lag constant for firn.
- icestorage (d) Storage constant for ice.
- icelag (h) Ice constant for SWE.
- TKMA (°C) Annual mean annual temperature of glacier.
- SWEAA (mm/yy) Annual mean accumulation of glacier SWE
- SWE_to_firn_Julian () convert any SWE to firn on Julian day. 0 inhibit transfer.
- Use_QnD () Use QnD if available. Basic module.
- tfactor (mm/d*/°C) degree day melt factor. Basic module.
- nfactor (mm*m^2/MJ*d) net radiation factor. Basic module.

Variable Inputs

- hru_tmean (obs) (°C)
- hru_tmax (obs) (°C)
- hru_tmin (obs) (°C)
- hru_umean (obs) (m/s)
- hru_rhmean (obs) (%)
- hru_Sun_Act (obs) (hr)
- net_rain (*) (mm/int) to simplify interfacing with an infiltration routine, glacier adds SWE, firn and ice melt to net_rain. Original net_rain contents are save in net_rain_org.
- QdroD (global) (MJ/m²)
- QdfoD (global) (MJ/m^2)

- SunMax (global) (h)
- SWE (*) (mm)
- snowmeltD (*) (mm/d)
- Qnsn_Var (W/m^2 day) net all-wave at snow surface. Variation glacier#1
- Pa (kPa) atmospheric pressure.

Useage.

Qn is calculated as in the original algorithm.

Over snow during melt;

Qn = -0.53+0.47*(QdroD+QdfoD)*(0.52+0.52*sunact/sunmax)*(1.0-Albedo)

An external observation QnD may be generated using modules long, NeedleLeaf and macro modules Slope_Qsi.mcr and Convert_Qsn_QnD.mcr.

glacier2

This module is defined in Classglacier and is an energy-budget snowmelt model loosely based on work done by (Gray and Landine, 1987) for the Canadian Prairies. However, it has variations allowing it to input the melt input energy from other CRHM modules.

Use with module "pbsm". Uses external melt model. Internal melt model for "ice" and "firn" is based upun module "ebsm" but by using variations 1 use "Qnsn_Var (variable)", variation 2 use "Qnsn (observation)" and with basic and variation 3 use "QnD (observation)" to operate using other CRHM modules.

Observations

- QnD (MJ/m² day) net radiation. Basic module glacier and glacier#3. Always optional.
- Qnsn (W/m² day) net all-wave at snow surface. Variation #2.

Variables

- firn_h (mm) depths of snow densification layers.
- firn_dens (1000 kg/m³) densities of snow densification layers.
- firn_yr (1000 kg/m^3) years to 'critical density' and age of firn.
- ice (m) water equivalent of ice. State variable.
- firn (m) water equivalent of all firn layers. State variable.
- firn_depth (m) depth of all firn layers. State variable.
- firn_init (m) Initial water equivalent of all firn layers.
- nfirn () number of layers used in firn densification.
- SWE (mm) snow water equivalent.
- SWEmelt (mm/d) daily snowmelt.
- SWEmelt_delayed (mm/d) daily snowmelt delayed by Clark.
- cumSWEmelt (mm/d) cummulative firn snowmelt.
- firnmelt (mm/d) daily firn melt.
- firnmelt_delayed (mm/d) daily firn melt delayed by Clark.
- cumfirnmelt (mm/d) cummulative firn melt.
- icemelt (mm/d) daily ice melt.
- icemelt_delayed (mm/d) daily ice melt delayed by Clark.

- cumicemelt (mm/) cummulative ice melt.
- infil (mm/int) depth of infiltration in time step, △t expressed as an average depth of water on an HRU.
- cuminfil (mm) cumulative infiltration expressed as an average depth of water on an HRU.
- runoff (mm/int) amount of daily runoff expressed as an equivalent depth (m³/m²).
- cumrunoff (mm) cummulative runoff.
- meltrunoff (mm/int) amount of interval melt runoff. .
- cummeltrunoff (mm) cumulative interval melt runoff.
- snowinfil (mm/int) not used. Satisfies module 'Netroute etc.
- LW_ebsm (mm/d) liquid water in snowpack. State variable.
- u_ebsm (MJ/m^2*d) snowpack energy deficit. State variable.
- Gmelt (MJ/m^2*d) Qmelt = Qn+Qh+Qe+Qp..
- Qn_ebsm (MJ/m^2*d) net radiation.
- Qh_ebsm (MJ/m^2*d) sensible heat.
- Qe_ebsm (MJ/m^2*d) latent heat.
- Qp_ebsm (MJ/m^2*d) input from rainfall.
- SWE_2firn (mm) end of summer SWE transferred to firn.
- SWE_2firn_dens (mm) end of summer density of SWE transferred to firn
- cumSWE_2firn (mm) cummulative SWE added to firn..
- LWmax (mm) -maximum liquid water in snowpack.
- net_rainD (mm/d) daily net rain calculated from net_rain.
- Qnsn_Acc (W/m^2) accumulator for Qnsn_Var/Qnsn. Variation glacier#1

Parameters

- delay_melt () inhibits any melt before this date. Only used when tfactor and nfactor are both zero. Default value not suitable for Southern Hemisphere
- basin_area (km^2) basin area.
- hru area (km^2) HRU area.
- firn_h_init (mm) initial depths of firn densification layers.
- firn_dens_init (1000 kg/m^3) initial densities of firn densification layers.
- ice_init (mm) Initial water equivalent of glacier ice.
- firn_Albedo () Initial glacier firn albedo.
- ice_Albedo () Initial glacier ice albedo.
- SWEstorage (d) Storage constant for SWE.

- SWElag (h) Lag constant for SWE.
- firnstorage (d) Storage constant for firn.
- firnlag (h) Lag constant for firn.
- icestorage (d) Storage constant for ice.
- icelag (h) Ice constant for SWE.
- TKMA (°C) Annual mean annual temperature of glacier.
- SWEAA (mm/yy) Annual mean accumulation of glacier SWE
- rho0 (kg/m³) zero depth intersection of ln(rho/(rhoice-rho)) could equal average density over first one or two meters of snow.
- SWE_to_infilK
- firn_to_infilK
- ice_to_infilK
- rain_to_infilK
- SWE_to_firn_Julian
- tfactor (mm/d*/°C) degree day melt factor. Basic module.
- nfactor (mm*m²/MJ*d) net radiation factor. Basic module.
- Use_QnD () Use QnD if available. Basic module.

Variable Inputs

- net_snow (*) (kg/m^2)
- net_rain (*) (kg/m^2)
- net_hru_drift (*) (kg/m^2)
- net_subl (*) (kg/m^2)
- hru_tmean (obs) (°C)
- hru_tmax (obs) (°C)
- hru_tmin (obs) (°C)
- hru_umean (obs) (m/s)
- hru_rhmean (obs) (%)
- hru_Sun_Act (obs) (hr)
- QdroD (global) (MJ/m^2)
- QdfoD (global) (MJ/m^2)
- SunMax (global) (h)
- Albedo (albedo) ()
- meltflag (albedo) ()
- Qnsn_Var (W/m^2 day) net all-wave at snow surface. Variation glacier#1

Useage.

tfactor and nfactor NOT equal to zero

For comparison there is an option to use the simplified energy budget method proposed by Kustas et al.(1994). In this degree-day algorithm the increased melting later in the season is simulated using a net radiation term.

Melt = tfactor $*t_{max}$ + nfactor $*Q_nD$, where the observation Q_nD is the net allwave radiation calculated from a module.

tfactor and nfactor equal to zero

If both parameters tfactor and nfactor are set equal to zero for an HRU, the module uses the original ebsm.f algorithm otherwise the simplified degree-day algorithm is used using the values set in tfactor and nfactor.

Observation QnD is used if available, otherwise Qn is calculated as in the original algorithm.

Over snow during melt;

Qn = -0.53+0.47*(QdroD+QdfoD)*(0.52+0.52*sunact/sunmax)*(1.0-Albedo)

An external observation QnD may be generated using modules long, NeedleLeaf and macro modules Slope_Qsi.mcr and Convert_Qsn_QnD.mcr.

Creating a Macro to dynamically change parameters at model run time.

Normally parameters have an intial value which is held constant throughout a model run. This is the correct behaviour for parameters like elevation, area etc. However, vegetation height is a variable during the growing season. It should be handled as a variable. This is easily accomplished using a *macro*. Two examples follow. The first illustrates how the vegetation height may read from an observation file and the second if the height versus time relationship is known.

1) Using an observation file holding vegetation heights.

The following macro reads the observation *Heights* and writes it every interval to the parameter *Ht*. The variable Ht_Var was created so that the heights could be displayed as there is no method of charting parameters in CRHM (assumed to be constants).

```
Ht_file
declreadobs,Heights, NOBS, "Vegetation height", (m)
declparam, Ht, NHRU,0.1, 0.01, 10.0, description, (m)
declvar, Ht_Var, NHRU, "vegetation height", (m)
command
    Ht[hh] = Heights[hh]
    Ht_Var[hh] = Ht[hh]
end
```

2) Programming vegetation heights.

The following macro is written assuming the crop grows continuously from a Julian date to a maximum value.

Ht_Change declparam, Ht, NHRU,0.1, 0.01, 100.0, description, (m) declparam, Ht_change, NHRU,0.1, 0.01, 1.0, description, (m) declparam, Ht_min, NHRU,0.1, 0.01, 3.0, description, (m) declparam, Ht_max, NHRU,0.1, 0.01, 3.0, description, (m) declparam, Ht_Julian, NHRU,195, 1, 366, description, () declvar, Ht_Var, NHRU, "vegetation height", (m)

```
declvar, Ht_Start, NHRU, "vegetation height", ()
command
  if(STEP == 1)
    Ht[hh] = Ht_min[hh]
  endif
  if(JULIAN >= Ht_Julian)
    Ht_Start[hh] = JULIAN
    if(LASTINT)
      Ht[hh] = Ht[hh] + Ht_change[hh]
    endif
    if( Ht[hh] > Ht_max[hh])
      Ht[hh] = Ht_max[hh]
    endif
  endif
  Ht_Var[hh] = Ht[hh]
end
```

Using the Height change macro modules.

After the macro code is added to the editor answer *YES* to the query asking if the modules should be loaded. The modules must be next added to the current model using the normal BUILD/CONSTRUCT screen. After adding the height macro module and during the build process, CRHM will ask if the module should be deleted, answer *NO*. This is necessary as CRHM is not able to determine ihow the module is used.

Units.

All observation, parameters and variables have assigned units. These are checked at run-time and any errors logged.

A sample observation file header follows.

Observation header file format

t 1 (°C) comment. e.g. air temperature.

rh 1 (%) comment. e.g. relative humidity.

u 1 (m/s) comment etc.

SunAct 1 (h)

ppt 1 (mm/d)

Qsi 1 (W/m^2)

Qso 1 (W/m^2)

Qn 1 (W/m^2)

\$Qsi mul(Qsi\t 277.8) unit conversion

\$Qso mul(Qso\t 277.8)

\$Qn mul(Qn\t 277.8)

\$ea ea(t\t rh) (kPa) vapour pressure

Note that there is one or more spaces or tabs between the number of observations and the units and after. The remainder of the line is treated as a comment.

Units must be enclosed in parentheses.

Case is significant in all units and multipliers..

° (degree) is not required but may be constructed using ALT key and Num Lock and 0176 or 0186 entered on the numeric key pad.

% (percentage) is ignored.

Multipliers.

Y	Yotta	1e+24
Z	Zetta	1e+21
E	Exa	1e+18
Р	Peta	1e+15
Т	Tera	1e+12
G	Giga	1e+9
Μ	Mega	1e+6
k	kilo	1e+3
K	Kilo	1e+3
h	hecto	1e+2
Η	Hecto	1e+2
D	Deka	1e+1
d	deci	1e-1
C	cent	1e-2
m	mili	1e-3
u	micro	1e-6
μ	micro	1e-6
n	nano	1e-9
р	pico	1e-12
f	femto	1e-15
a	atto	1e-18
Z	zepto	1 e- 21

Units.

А	Ampere	1	
Bq	becquerel	1	l/s
Btu	InternationalTableBtu	1055.05585262	Kg*m^2/s^2
С	Coulomb	1	A*s
Ci	Curie	3.7*1010	l/s
F	Farad	1	A^2*s^4/Kg*m^2
Fdy	Faraday	96487	A*s
Gy	Gray	1	m^2/s^2
Н	Henry	1	Kg*m^2/A^2*s^2
Hz	Hertz	1	s^-1
J	Joule	1	Kg*m^2/s^2
K	Kelvin	1	"),
Knot	NauticalMilesPerHour	463.0/900.0	m/s
Ν	Newton	1	Kg*m/s^2
Ohm	Ohm	1	Kg*m^2/A^2*s^3
OzUK	UKFluidOunce	2.8413075e-5	m^3
Р	Poise	0.1	Kg/m*s
Ра	Pascal	1	Kg/m*s
Pdl	Poundal	0.13825495376	Kg*m/s^2
Pica	Pica	1.0/72.0	in
R	Roentgen	0.000258	A*s/Kg
S	Siemens	1	A^2*s^3/Kg*m^2
St	Stokes	0.0001	m^2/s
Sv	Sievert	1	m^2/s^2
Т	Tesla	1	Kg/A*s^2
U	UnifiedAtomicMass	1.6605402e-27	Kg
V	Volt	1	Kg*m^2/A*s^2

W	Watt	1	Kg*m^2/s^3
Wb	Weber	1	Kg*m^2/A*s^2
acre	Acre	4046.87260987	m^2
arcmin	ArcMinute	2.9088820866e-4	r
arcs	ArcSecond	4.848136111e-6	r
atm	Attmosphere	101325	Kg/m*s^2
au	AstronomicalUnit	1.495979*1011	m
b	Barn	1e-28	m^2
bar	Bar	100000	Kg/m*s^2
bbl	Barrel	0.158987294928	m^3
bu	Bushel	0.03523907	m^3
С	LightSpeed	299792458	m/s
cal	Calorie	4.1868	Kg*m^2/s^2
cd	Candela	1	
chain	Chain	20.1168402337	m
ct	Carat	0.0002	Kg
cu	USCup	2.365882365e-4	m^3
d	day	86400	S
dyn	Dyne	0.00001	Kg*m/s^2
eV	ElectronVolt	1.60217733e-19	Kg*m^2/s^2
erg	Erg	0.0000001	Kg*m^2/s^2
fath	Fathom	1.82880365761	m
fbm	BoardFoot	0.002359737216	m^3
fc	Footcandle	10.7639104167	cd*sr/m^2
fermi	Fermi	1e-15	m
flam	Footlambert	3.42625909964	cd/m^2
ft	InternationalFoot	0.3048	m
ftUS	SurveyFoot	0.304800609601	m
g	Gram	1	
ga	StandardFreefall	9.80665	m/s^2
gal	USGallon	0.003785411784	m^3

galC	CanadianGallon	0.00454609	m^3
galUK	UKGallon	0.004546092	m^3
gf	GramForce	0.00980665	Kg*m/s^2
grad	Grade	1.57079632679e-2	r
grain	Grain	0.00006479891	Kg
h	Hour	3600	S
ha	Hectare	10000	m^2
hp	horsepower	745.699871582	Kg*m^2/s^2
in	Inch	0.0254	m
inH2O	InchesOfWater	248.84	Kg/m*s^2
inHg	InchesOfMercury	3386.38815789	Kg/m*s^2
int	CRHM time step	3600	S
kip	KilopoundForce	4448.22161526	Kg*m/s^2
kph	KilometersPerHour	5.0/18.0	m/s
1	Liter	0.001	m^3
lam	Lambert	3183.09886184	cd/m^2
lb	AvoirdupoisPound	0.45359267	Kg
lbf	PoundForce	4.44822161526	Kg*m/s^2
lbt	TroyPound	0.3732417216	Kg
lm	Lumen	1	cd*sr
lx	Lux	1	cd*sr/m^2
lyr	LightYear	9.46052840488*1015	m
m	Meter	1	
mho	Mho	1	A^2*s^2/Kg*m^2
mi	InternationalMile	1609.344	m
miUS	USStatuteMile	1609.34721869	m
mil	Mil	0.0000254	m
min	Minute	60	S
mmHg	MilimeterOfMercury	133.322368421	Kg/m*s^2
mol	Mole	1	

mph	MilesPerHour	0.44704	m/s
nmi	NauticalMile	1852	m
OZ	Ounce	0.028349523125	Kg
ozfl	USFluidOunce	2.95735295625e-5	m^3
ozt	TroyOunce	0.0311034768	Kg
рс	Parsec	3.08567818585106	m
ph	Phot	10000	cd*sr/m^2
pk	Peck	0.0088097675	m^3
psi	PoundsPerSquareInch	6894.75729317	Kg/m*s^2
pt	Pint	0.000473176473	m^3
qt	Quart	0.000946352946	m^3
r	Radian	1	
rad	Rad	0.01	m^2/s^2
rd	Rod	5.02921005842	m
rem	Rem	0.01	m^2/s^2
S	Second	1	
sb	Stilb	10000	cd/m^2
slug	Slug	14.5939029372	Kg
sr	Steradian	1	
st	Stere	1	m^3
t	MetricTon	1000	Kg
tbsp	Tablespoon	1.47867647813e-5	m^3
therm	EECTherm	105506000	Kg*m^2/s^2
ton	ShortTon	907.18474	Kg
tonUK	UKLongTon	1016.0469088	Kg
torr	Torr	133.322368421	Kg/m^2
tsp	Teaspoon	4.92892159375e-6	m^3
yd	InternationalYard	0.9144	m
yr	Year	31556925.9747	S
0	Degree	1.74532925199e-2	r

°C	DegreeCelsius	1.0	K
°F	DegreesFahrenheit	1.0/1.8	К
°R	DegreesRankine	1.0/1.8	K
μ	Micron	1e-6	m
Å	Angstrom	1e-10	m