Glacier mass balance simulation from point measurements

enrico.mattea@unifr.ch

March 2021

Chapter 1

Overview

The purpose of this software package is to model the distributed annual mass balance of a glacier. The simulation is driven by a daily meteorological series of temperature and precipitation, and is constrained by point measurements of melt (and optionally of snow accumulation).

Chapter 2

Model description

The simulation is performed on a **gridded domain**, defined by a digital height model (DHM). The mass balance model is driven by a **daily meteorological series** of temperature and precipitation: starting from an initial condition of snow-water equivalent (SWE), the model computes daily accumulation and melt for each grid cell, keeping track of the accumulated mass balance. **Each year is simulated individually**: the simulated period for a year is defined to include the whole observation period of the stakes measured on that year, as well as the whole hydrological year (1 October-30 September) and also possibly a user-defined period. So in general the simulated period for a single year will be more than 365 days, and in a multi-year simulation there will be some overlap (this is not an issue since the final mass balance is extracted at the correct dates). The simulation of different years can use different input height models and glacier outlines: this is useful if the glacier shape or topography change during the modeled years. There is also an option to automatically **interpolate between elevation grids** of different years.

The **initial snow condition** for a year is computed from topography (elevation, slope, curvature), from a user-defined snow line altitude and elevation gradient, from an avalanche simulation and (if available) from the winter snow probes. From the second modeled year there is the option to use the previous model output as starting condition.

Distributed **daily melt** is computed using an enhanced temperature index (degree-day) model (Hock, 1999), considering both mean daily air temperature and potential solar radiation. Different radiation factors are used for snow, bare ice and debris-covered ice. Additionally, there is the option to use a spatially variable melt factor for ice, to account for darker ice towards the glacier terminus. Distributed **daily accumulation** is derived from precipitation, using a rain/snow temperature threshold and a distribution grid similar to the one for the initial snow condition.

The model features a **process-based avalanche routine** (Gruber, 2007) to redistribute the snow mass. Avalanches are simulated at user-defined fixed dates; they can change the net amount of snow on the glacier surface (this is why the entire grid is simulated and not only the glaciated cells). Elevation grids are **automatically processed** to ensure a **correct hydrologic flow** of the avalanche: all grid sinks are filled using 4-connectivity (that is, from each grid cell there is a strictly descending path to the grid border going through the four surrounding neighbors). Still, for more control it can be a good idea to use elevation grids where the sinks are already filled.

The model for each year is automatically run several times to **optimize the parameters**, in order to find the best fit to the measurements. This optimization works **by comparing the stake measurements to the modeled mass balance** at the location of each stake (bilinearly filtered from the grid). The comparison is performed over the exact observation period of each stake (so multiple observations during a year can be used). The parameters are tuned to **cancel the mean stake bias** (the RMS error is not considered). If winter measurements are available (snow probes), the model is first run over the winter period only, to optimize a factor of **precipitation correction** against the winter stake measurements. Then the model is run over the entire period (longer than a year, as described above) to optimize the **melt factors** against the annual stake measurements. The model also supports **snow depth measurements with an unknown starting date**: the starting date is automatically set to the date of the previous annual mass balance minimum. This can be used to include summer measurements from snow pits and snow probes. If the measured stakes are not uniformly distributed over the glacier, the optimization can be biased towards a certain glacier area. To mitigate this, the model can **cluster together stakes** which are closer than a distance threshold, treating them as a single stake.

After optimization over the global bias, the model performs a **final local correction of the simulated mass balance** within user-defined **elevation bands**: for each elevation band it computes the model bias compared to the stakes in that band, and corrects the model output accordingly (with a linear interpolation between bands, for a smooth result). This accounts for differences between the model result and the contour-line method.

Chapter 3

Installation

The model mostly uses the free programming language R; for performance reasons one routine (the process-based avalanche simulation) is written in C++.

The model was developed using R version 3.6.3. It has not been tested under other versions. To run the model you need at least:

- the **R language interpreter** (installation instructions)
- (not mandatory, but strongly recommended) the **RStudio IDE** (download page)
- **a code compiler**, to install R packages and compile the avalanche module: see here
- the **R packages** used by the project. These are listed in Table 3.1; most of them also depend on additional packages which are automatically installed as required.

  Packages from the *Comprehensive R Archive Network* (CRAN) can be installed with

  ```
  install.packages("<package name>")
  ```

  GitHub packages can be installed with

  ```
  remotes::install_github("<repository>/<package>")
  ```

  Additional information on installing R packages can be found here.

Name	**Repository**	**Purpose**
cowplot	CRAN	Align plots
ggplot2	CRAN	Base plotting library
ggpubr	CRAN	Multi-page PDF
gstat	CRAN	Spatial interpolation of snow probes
metR	CRAN	*geom_text_contour()*
raster	CRAN	Grid manipulation
Rcpp	CRAN	Avalanche function implemented in C++
remotes	CRAN	*install_github()*
reshape2	CRAN	*melt()*
Rfast	CRAN	*rowSort()*
scales	CRAN	*rescale()*
stringr	CRAN	*str_split()*
sf	CRAN	*st_read()*
sp	CRAN	*SpatialPolygons()*, for glacier outlines
spatialEco	CRAN	*curvature()*
topmodel	CRAN	*sinkfill()*
timeSeries	CRAN	*interpNA()*
ggpattern	GitHub	Pattern as histogram fill

Table 3.1: R packages explicitly loaded by the project

Chapter 4

Workflow

4.1 Input files

This section describes the model input preparation. The required input data consist of:

- one or more elevation grids (rasters)
- one or more grids of surface type
- 365 daily radiation grids
- one or more glacier outlines in vector format
- a text file with the daily meteorological series
- one or two files with the annual (and optionally winter) point measurements of mass balance

All grids (elevation, radiation and surface type) should have the **same coordinates system, extent and resolution**. Grids of a same variable should also use the same file format and follow a standard naming convention (see Sect. 4.2). Supported **file formats** can be listed using R command `rgdal::gdalDrivers()`. Common choices include GeoTIFF and ESRI/ASCII grid. The **grid extent** should be wide enough to include a margin of some row/columns around the glacier. Ideally, the whole water catchment of the glacier should be included (for avalanches which can reach the glacier from outside). **Grids should not have any missing (NA/NaN/nodata) value.**

If available, **multiple grids** (max one per year) can be provided in order to run the simulation on up-to-date input data: for example repeated digital elevation models. The grid years do not have to match: one could supply several elevation grids but just a single file for the surface type.

For **elevation** there is the option to linearly interpolate the grids for the missing years (else the grid which is closest in time to the modeled year will be selected). The model will process the elevation grids to obtain a hydrologically correct map, with no sinks. For more control you can provide elevation grids where the sinks have already been filled. Be aware that the linear interpolation between two sink-filled grids could still produce sinks, which will be removed by the model.

Grids of **surface type** use the codes reported in Table 4.1. Note that the glacier cells are derived from the vector outline (see below): therefore there shall be no grid cell with code 4 (non-glacier area) inside the glacier outline, because the simulated mass balance of this cell would be wrong (melt would stop as soon as there is no more snow on the cell). **To produce an accurate grid of surface type**, it is best to start from a grid of non-glacier (value 4 everywhere), then use the glacier outline as a mask to put the value 0 over the glacier, and then draw other vector masks to change the value of firn and debris cover. Under QGIS, the relevant tools are

`Layer → Create Layer → New Temporary Scratch Layer`
`Raster → Conversion → Rasterize`
`Raster → Raster Calculator`.

The grids of **daily radiation** are used within the melt model. Radiation should be expressed in W m^{-2}. You should provide just 365 daily grids (inter-annual variability of radiation is ignored). On leap years, the model replicates

Code	Meaning
0	Bare ice
1	Firn
4	Non-glacier areas
5	Debris-covered ice

Table 4.1: surface type codes

the 365$^{\text{th}}$ day twice. The grids can be generated using any tool. Under the *utils/* directory there is an R script to generate them using the SAGA GIS command line interface (explained here; installation instructions here). Some minor adaptations may be needed to run the script on Windows and Mac OS X systems.

Glacier outlines in vector format are used to select the grid cells which contribute to mass balance, and also to plot the output. Like the grids of elevation and surface type, you can supply several outlines which correspond to different years. Outlines can use either the ESRI shapefile format (.shp) or the XYZN format.

The **meteorological series** for the whole simulation should be provided as a single text file with 5 columns (space- or tab-separated): year, day of year (1-366), hour (ignored), temperature (in °C) and precipitation (in mm w.e.). The model will select the appropriate period for each modeled year. The series should be long enough to cover the full simulation period, which is usually longer than the period of stake observations as it includes the hydrological year.

Mass balance observations should be provided as one text file for annual measurements, and optionally another for winter measurements. The files should have 8 columns (space- or tab-separated): point name, start of observation period, end of observation period, X coordinate, Y coordinate, Z coordinate (altitude), stake measurement in cm and density in g cm^{-3}. The observation periods should be specified as DD.MM.YYYY, and the coordinates should use the same system as the grids. One **special value for the start of the measurement period** is **NA**: it is interpreted by the model as **"end of the previous melting season"** and should be used for snow pits and snow probes, which measure down to the previous summer surface (see Chapter 2). If the stake measurement is known already as water-equivalent (and not stake height + density) you can simply set the density to 1.

4.2 Model parameters

The simulation is controlled by two groups of parameters: one group which is fixed for the whole run (over all years), and one which changes each year (annual parameters). Fixed parameters are set in file *set_params.R*; annual parameters should be provided as text files (one per year).

Describe format of annual parameters file! Also, which parameters are most important?

4.3 Model run

Simply run the main.R file. If you run it several times (for example adjusting some parameters), you may want to save a boot file so that data loading is much faster the next time (if you change anything in the data then you will have to re-load all data and re-create a new boot file).

4.4 Output files

Describe what the model generates.

Chapter 5

Code architecture

Functions and procedures.

Bibliography

Gruber, S.: A mass-conserving fast algorithm to parameterize gravitational transport and deposition using digital elevation models, Water Resources Research, 43, https://doi.org/10.1029/2006WR004868, URL http://doi.wiley.com/10.1029/2006WR004868, 2007.

Hock, R.: A distributed temperature-index ice- and snowmelt model including potential direct solar radiation, Journal of Glaciology, 45, 101–111, https://doi.org/10.3189/S0022143000003087, URL https://www.cambridge.org/core/product/identifier/S0022143000003087/type/journal_article, 1999.