pySnowClim: fast process-based snow modeling in Python

Aranildo R. Lima [1¶], **Abby C. Lute** [2], and **Rajesh R. Shrestha** [1]

1 Climate Research Division, Environment and Climate Change Canada, Victoria, BC, Canada **2** Woodwell Climate Research Center, Falmouth, MA 02540, USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Jack Atkinson

Submitted: 08 October 2025
Published: unpublished

Summary

pySnowClim is an open-source Python implementation of the process-based SnowClim model for simulating snow dynamics including accumulation and melt processes. The model achieves a balance between representing important physical processes, such as energy balance calculations, snow density evolution, albedo dynamics, and phase change processes, while simplifying other components to maintain computationally efficiency, delivering consistent snowpack simulations across diverse climatic conditions. These characteristics make pySnowClim ideal for large-scale regional to continental studies, studies requiring $< 1km$ spatial resolution such as those in complex terrain, and studies evaluating the impact of alternate climate scenarios on snowpack.

Statement of need

Snow is a critical component of the global water cycle. The accurate simulation of snowpack dynamics is not only essential for water resource management, but also for flood forecasting, ecological studies, and climate change impact assessments (Caretta et al., 2022).

The target audience of `pySnowClim` includes hydrologists, climatologists, ecologists, water resource managers, and students who need reliable snow modeling capabilities for applications such as:

- **Research**: Detailed energy balance studies and process investigations
- **Operations**: Water resource forecasting and management
- **Education**: Teaching snow physics and energy balance concepts
- **Climate Studies**: Long-term snow evolution under changing conditions
- **Adaptation Planning**: Anticipating and planning adaptations to impacts of warming on snow-dependent species, ecosystems, and communities

In addition, the model can be used on different:

- **Spatial Scales**: Point locations to continental domains
- **Temporal Scales**: Sub-daily to daily timesteps, multi-decadal simulations
- **Environments**: Diverse snow climates from maritime to continental

State of the field

Many current snow models are either computationally efficient but only represent physical processes to a very limited extent (e.g. temperature index models), or represent many important physical processes but are too computational burdensome for large-scale high-resolution applications (e.g. most process-based models). In addition, many are propriety, not easily accessible, or difficult to integrate with modern scientific workflows (Garen & Marks, 2005;

Ikeda et al., 2021; Liston & Elder, 2006; Walter et al., 2005; Wrzesien et al., 2018). `pySnowClim` addresses some of these problems by offering a flexible, efficient, and open alternative with a good balance between representing physical processes and usability.

Software design

The model architecture of `pySnowClim` package follows object-oriented design principles with clear separation between forcing data handling, parameter management, core physics calculations, and output generation. `pySnowClim` leverages NumPy's vectorized operations for processing multiple grid points simultaneously, enhancing computational efficiency. It also uses xarray and pandas for more efficient data manipulation and NetCDF I/O operations. The package includes a comprehensive testing framework, featuring unit tests for individual physics functions and integration tests for complete workflows. `pySnowClim` also provides extensive documentation (https://abbylute.github.io/pySnowClim), including API references, example datasets, and validation against observations from a snow monitoring site. Model outputs include standard snow variables (SWE, depth, density, melt, albedo, temperature) as well as detailed energy budget components.

Research impact statement

There are several key improvements of `pySnowClim` compared to the original MATLAB-based SnowClim model (Lute et al., 2022). A Python package makes the model more accessible to a wider scientific and practical audience. Python is widely adopted in earth science and data science communities, and a well-supported package enables easier integration, reproducibility, and further development. An example is the current implementation of coupling between `pySnowClim` and the Community Water Model (CWatM) (Burek et al., 2020) (https://github.com/iiasa/CWatM). In addition, to make the model more realistic, a new functionality was added to reduce excessive snow accumulation (i.e. snow towers) using an optional radiation enhancement.

The model

`pySnowClim` employs the fundamental principles of mass and energy conservation as its core framework. The model requires meteorological forcing data including temperature, precipitation, shortwave radiation, longwave radiation, wind speed, humidity, and pressure to simulate critical snow variables such as snow water equivalent (SWE), snow depth, snow density, snow melt, snowpack liquid water content, albedo, and energy fluxes. Figure 1 shows the snow model conceptual diagram.

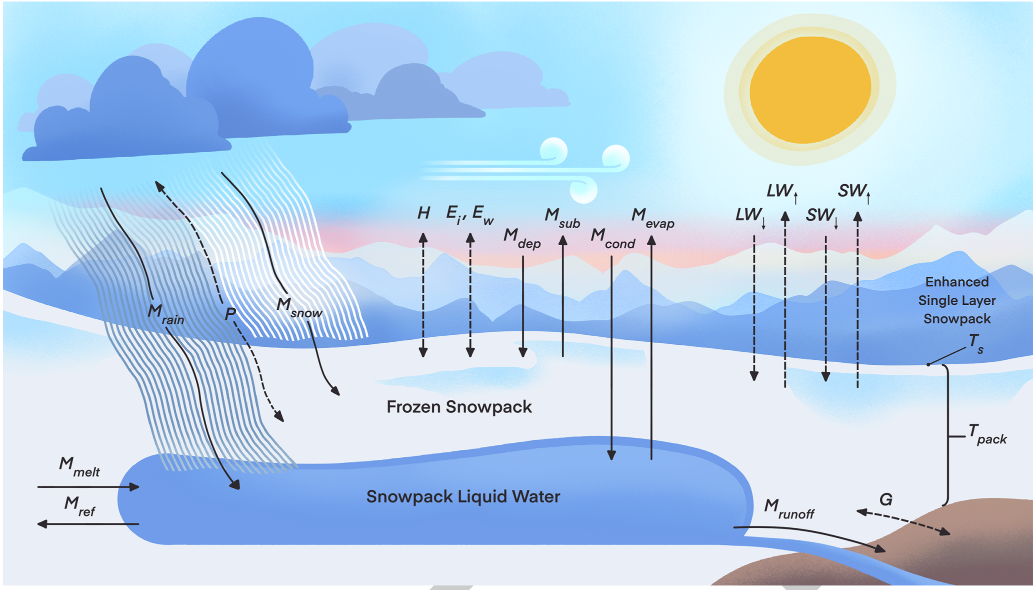

Figure 1: Snow model conceptual diagram. Solid black arrows indicate mass fluxes and dashed black arrows indicate energy fluxes. T_s is snow surface temperature and T_{pack} is the temperature of the snowpack.

pySnowClim calculates the net energy flux to the snow surface accounting for shortwave and longwave radiation, sensible and latent heat fluxes with stability corrections, ground heat flux, and precipitation heat flux Equation 1. The model is built around the surface energy balance equation:

$$Q_{net} = SW_{down} - SW_{up} + LW_{down} - LW_{up} + H + E_i + E_w + G + P \tag{1}$$

Where SW denotes shortwave radiation fluxes, LW denotes longwave radiation fluxes, H is the sensible heat flux, E is the latent heat flux of ice (i) and water (w), G is the ground heat flux, and P is the advected heat flux from precipitation.

The model tracks snow accumulation, sublimation and evaporation processes, snowmelt generation, and liquid water movement through the snowpack with mass conservation Equation 2. Mass balance of the solid (M_s) and liquid (M_l) portions of the snowpack are governed by:

$$M_s = M_{snow} + M_{ref} - M_{melt} + M_{dep} - M_{sub} \tag{2}$$

$$M_l = M_{rain} - M_{ref} + M_{melt} - M_{runoff} + M_{cond} - M_{evap} \tag{3}$$

Where M_{snow} is the mass of new snowfall, M_{ref} is the mass of the snowpack liquid water that has been refrozen, M_{melt} is the mass of snow that has melted, M_{dep} is the mass of deposition, M_{sub} is the mass of sublimation, M_{rain} is the mass of rain added to the snowpack, M_{runoff} is the mass of liquid water that has left the snowpack as runoff, M_{cond} is the mass of condensation, and M_{evap} is the mass of evaporation.

Other key model components include precipitation phase determination using bivariate logistic regression (Jennings et al., 2018), fresh snow density calculation based on temperature (Anderson, 1976), snow density evolution through compaction processes, liquid water retention and drainage. pySnowClim has three different albedo scheme implementations including aging effects, melting conditions, and seasonal variations with options for different complexity levels (Essery et al., 2013; Hamman et al., 2018; Liang et al., 1994; Tarboton & Luce, 1996).

AI usage disclosure

In preparing this work, the authors used ChatGPT (OpenAI) and Claude (Anthropic) to assist with the initial MATLAB-to-Python translation and generation of code documentation. The unit tests of the package were carried out by using Claude. After using these tools, the authors thoroughly reviewed, edited and refined the content, and take full responsibility for the content of this publication. No generative AI tools were used in the development of the writing of this manuscript.

Acknowledgements

We would like to thank Dr. Alex Cannon and Dr. Narayan Shrestha for their constructive comments and suggestions. In addition, we thank Lincoln Lute for providing the conceptual diagram of the snow model.

References

Anderson, E. A. (1976). *A point energy and mass balance model of a snow cover* (No. 19; NOAA Technical Report NWS). https://repository.library.noaa.gov/view/noaa/6392

Burek, P., Satoh, Y., Kahil, T., Tang, T., Greve, P., Smilovic, M., Guillaumot, L., Zhao, F., & Wada, Y. (2020). Development of the community water model (CWatM v1.04) – a high-resolution hydrological model for global and regional assessment of integrated water resources management. *Geoscientific Model Development*, *13*(7), 3267–3298. https://doi.org/10.5194/gmd-13-3267-2020

Caretta, M. A., Mukherji, A., Arfanuzzaman, M., Betts, R. A., Gelfan, A., Hirabayashi, Y., Lissner, T. K., Liu, J., Lopez Gunn, E., Morgan, R., Mwanga, S., & Supratid, S. (2022). Water. In H.-O. P"ortner, D. C. Roberts, M. Tignor, E. S. Poloczanska, K. Mintenbeck, A. Alegr'ia, M. Craig, S. Langsdorf, S. L"oschke, V. M"oller, A. Okem, & B. Rama (Eds.), *Climate change 2022: Impacts, adaptation and vulnerability. Contribution of working group II to the sixth assessment report of the intergovernmental panel on climate change* (pp. 551–712). Cambridge University Press. https://doi.org/10.1017/9781009325844.006

Essery, R., Morin, S., Lejeune, Y., & B Ménard, C. (2013). A comparison of 1701 snow models using observations from an alpine site. *Advances in Water Resources*, *55*, 131–148. https://doi.org/10.1016/j.advwatres.2012.07.013

Garen, D. C., & Marks, D. (2005). Spatially distributed energy balance snowmelt modelling in a mountainous river basin: Estimation of meteorological inputs and verification of model results. *Journal of Hydrology*, *315*, 126–153. https://doi.org/10.1016/j.jhydrol.2005.03.026

Hamman, J. J., Nijssen, B., Bohn, T. J., Gergel, D. R., & Mao, Y. (2018). The variable infiltration capacity model version 5 (VIC-5): Infrastructure improvements for new applications and reproducibility. *Geoscientific Model Development*, *11*(8), 3481–3496. https://doi.org/10.5194/gmd-11-3481-2018

Ikeda, K., Rasmussen, R., Liu, C., & al., et. (2021). Snowfall and snowpack in the western u.s. As captured by convection permitting climate simulations: Current climate and pseudo global warming future climate. *Climate Dynamics*, *57*, 2191–2215. https://doi.org/10.1007/s00382-021-05805-w

Jennings, K. S., Winchell, T. S., & Livneh, B. (2018). Spatial variation of the rain–snow temperature threshold across the northern hemisphere. *Nature Communications*, *9*, 1148. https://doi.org/10.1038/s41467-018-03629-7

Liang, X., Lettenmaier, D. P., Wood, E. F., & Burges, S. J. (1994). A simple hydrologically based model of land surface water and energy fluxes for general circulation models. *Journal of Geophysical Research: Atmospheres*, *99*(D7), 14415–14428. https://doi.org/10.1029/94JD00483

Liston, G. E., & Elder, K. (2006). A distributed snow-evolution modeling system (SnowModel). *Journal of Hydrometeorology*, *7*(6), 1259–1276. https://doi.org/10.1175/JHM548.1

Lute, A. C., Abatzoglou, J., & Link, T. (2022). SnowClim v1.0: High-resolution snow model and data for the western United States. *Geoscientific Model Development*, *15*(13), 5045–5071. https://doi.org/10.5194/gmd-15-5045-2022

Tarboton, D. G., & Luce, C. H. (1996). *Utah energy balance snow accumulation and melt model (UEB): Computer model technical description and user guide*. Utah Water Research Laboratory; USDA Forest Service Rocky Mountain Research Station. https://hydrology.usu.edu/dtarb/snow/snowrep.pdf

Walter, M. T., Brooks, E. S., McCool, D. K., King, L. G., Molnau, M., & Boll, J. (2005). Process-based snowmelt modeling: Does it require more input data than temperature-index modeling? *Journal of Hydrology*, *300*(1-4), 65–75. https://doi.org/10.1016/j.jhydrol.2004.05.002

Wrzesien, M. L., Durand, M. T., Pavelsky, T. M., Kapnick, S. B., Zhang, Y., Guo, J., & Shum, C. K. (2018). A new estimate of north american mountain snow accumulation from regional climate model simulations. *Geophysical Research Letters*, *45*, 1423–1432. https://doi.org/10.1002/2017GL076664