We thank the reviewer for the constructive comments and suggestions. The comments are very helpful and will certainly strengthen the quality of the manuscript. Our response to the review can be found in the attached document.

R: Referee's comment
A: Author's response
C: Proposed changes in the manuscript
Blue letters: Suggested changes in the text

REVIEWER 1:

General comments

R: This paper describes a distributed surface energy and mass balance model coded in Python and available as open source on github. The paper describes in much detail the physics included in the model. The paper is well written and concise, although sometimes a bit too concise, see remarks below.

My main concern with this manuscript is that in my view it does not present anything new. There are several distributed energy and mass balance models available, some are more sophisticated than this one, some less, and at least one of them is also available as open source on github. The model itself is also not new, there are several publications with an earlier version of this model (Huintjes et al. 2014 and 2015), and the model physics in general is used in the other models as well and is already described in similar detail in other studies. I am not sure whether there are more of these type of models programmed in python, but that does not seem the key point here. Thus, what makes this model special or new to warrant publication?

A: Thank you for the thoughts. In fact, there are several distributed glacier mass balance models of varying complexity. The highest complexity is certainly reached by snow cover models (e.g. Snowpack, Crocus, etc.), some of which are freely accessible and actively maintained. COSIPY is a new edition of the obsolete Matlab version COSIMA (which was also developed by the first author). The differences between the models are, apart from the programming language, especially the model structure. This includes the discretization of the computational grid, the selection and implementation of the parameterizations, input/output routines, parallelization, etc. In addition, great emphasis was put on the documentation and readability of the code to offer other scientists the opportunity to actively participate in the further development. A documentation of this kind was not available in COSIMA. Since the differences between the versions are essential, and COSIPY already benefits from a relatively large community, we think that a citable article describing the model is needed. In our opinion, the GMD Journal is the appropriate platform for the description of new geoscientific models such as COSIPY.

In summary, here are the main points where COSIPY differs from other glacier mass balance models:

- is completely written in Python, modular and object-oriented
- completely based on open-source libraries
- has a readthedocs documentation (which is still in the development phase)
- parameterizations can be easily extended or modified by the user
- NetCDF IO
- easy integration of Weather Research and Forecast (WRF) forcing

- adapted for distributed glacier mass balances simulations; it needs to be pointed out in this regard that COSIPY is not a snow cover model
- has a community platform (Slack) and code is actively maintained
- new git commits are automatically tested via travis and codecov
- each model version gets a DOI
- has a restart option for operational applications

R: **It should be made much more clear what is new about this (see above).**

A: We have tried to highlight the differences (see comment above) between COSIPY and other models at several points in the text, such as:

P2L18-p3L6: "… *Ideally, a platform should (i) be continuously maintained, (ii) provide newly developed parameterisations, (iii) compile different model subversions developed for specific research needs, (iv) be easily extensible and (v) be well documented and readable. Here we present an open-source coupled snowpack and ice surface energy and mass balance model in Python (COSIPY) designed to meet these requirements. The structure is based on the predecessor model COSIMA (COupled Snowpack and Ice surface energy and MAss balance model, Huintjes et al., 2015). COSIPY provides a lean, flexible and user-friendly framework for modelling distributed snow and glacier mass changes. The framework consists of a computational core that forms the runtime environment and handles initialization, input-output (IO) routines, parallelization, and the grid and data structures. In most cases, the runtime environment does not require any changes by the user. Physical processes and parameterisations are handled separately by modules. The modules can be easily modified or extended to meet the needs of the end user. This structure provides maximum flexibility without worrying about internal numerical issues. The model is provided on a freely accessible git repository (https://github.com/cryotools/cosipy) and can be used for non-profit purposes. Scientists can actively participate in extending and improving the model code*".

We will expand this paragraph with additional information about the model structure and special features of COSIPY (see answer above).

C: (proposed changes to the original text are provided in **bold blue** letters) "… Ideally, a platform should (i) be continuously maintained, (ii) provide newly developed parameterisations, (iii) compile different model subversions developed for specific research needs, (iv) be easily extensible and (v) be well documented and readable. Here we present an open source coupled snowpack and ice surface energy and mass balance model in Python (COSIPY), which meets these requirements. The structure is based on the predecessor model COSIMA (COupled Snowpack and Ice surface energy and MAss balance model, Huintjes et al., 2015). COSIPY provides a lean, flexible and user-friendly framework for modelling distributed snow and glacier mass changes. The framework consists of a computing kernel that forms the runtime environment and handles initialization, input-output (IO) routines, parallelization, and grid and data structures. In most cases, the runtime environment does not require any changes by the user. **To increase the user friendliness, additional features are available to the user, such as a restart option for operational applications and automatic comparison between simulation and ablation stakes. The features will be further refined during the development phase.** Physical processes and parameterizations are handled separately by modules. The modules can easily be modified or extended to meet the needs of the end-user. This structure offers maximum flexibility without worrying about internal numerical issues. The model **is completely based on open-source libraries and** is provided on a freely accessible git repository (https://github.com/cryotools/cosipy) **for non-profit purposes.** Scientists can actively participate in extending and improving the model code. **Changes to the code are automatically tested with Travis CI (www.travis-ci.org) when uploaded to the**

repository. It is planned to publish updates in regular intervals. To make working with COSIPY easier, a community platform (https://cosipy.slack.com) has been set up in addition to a detailed readthedocs documentation (https://cosipy.readthedocs.io/en/latest), allowing users and developers to exchange experiences, report bugs and communicate needs.

S: Done

R: **In my experience it is often not so much the model formulation and running of it that is a problem, but the preparation of the input data. In this manuscript there is almost no information on how the input data is prepared and how it is distributed over the grid. Is this provided for in this package or should the user do that him/herself? And if it is included, how is it done? Make clear what the user is suposed to do him/herself and what is included.**

A: We agree with the reviewer and identify the data pre-processing as one of the most important steps in the modelling process, but the pre-processing differs from case to case and the user's needs. For this reason, COSIPY does not provide any standard pre-processing routines and it is up to the user to prepare or spatially interpolate the data. However, we do provide example scripts that illustrate and facilitate the data preparation workflow for the user. An example is given in the online documentation (https://cosipy.readthedocs.io/en/latest/Documentation.html#quick-tutorial). The example shows how the input dataset can be generated from automatic weather station data and a given digital elevation model. The example scripts use simple lapse rates for temperature, precipitation and humidity for the interpolation. The wind speed, cloud cover and longwave radiation is assumed to be constant over the domain. For the interpolation of the radiation the radiation model of Wohlfahrt et al (2016) is used (doi: 10.1016/j.agrformet.2016.05.0120).

C: Chapter 4.1 deals explicitly with the input/output and refers to the corresponding website. In this chapter it says:

"*The model is driven by meteorological data that must be provided in a corresponding NetCDF file (see https://cosipy.readthedocs.io/en/latest/Ressources.html). Input parameters include atmospheric pressure, air temperature, cloud cover fraction, relative humidity, incoming shortwave radiation, total precipitation and wind velocity. Optional snowfall and incoming longwave radiation can be used as forcing parameters. In addition to meteorological parameters, COSIPY requires static information such as topographic parameters and a glacier mask.* ***Example workflows for creating and converting static and meteorological data into the required NetCDF input is included in the source code (*****https://cosipy.readthedocs.io/en/latest/Documentation.html#quick-tutorial)*****.*** *Besides the standard output variables, there is also the possibility to store vertical snow profile information (not recommended for distributed simulations). To reduce the amount of data, the users can specify which of the output variables will be stored*."

We will change the phrase 'Various tools are available ...' to 'Example workflows for creating and converting static and meteorological data into the required NetCDF input is included in the source code (https://cosipy.readthedocs.io/en/latest/Documentation.html#quick-tutorial)'.

S: Done

R: **Other information I am missing is on initial conditions, tuning and spin up. What procedure do you use? Is this also something provided for in the package or has the user do this him/herself?**

A: Simulations depend on the initial conditions and the spin-up time. To ensure maximum flexibility, these must be specified by the user by choosing an adequate simulation period and initial conditions. As with all models, the model is calibrated by adjusting the model parameters and constants. The user has the possibility to adjust all parameters and constants of the parameterization in the configuration file. Which metrics users want to use for the evaluation depends on their specific application. By default, COSIPY automatically calculates the root mean squared error between the simulation and ablation measurements, if the measurements are provided.

C: We will rename the title of Section 4.1 to '**Input/Output** (**IO) and initial condition**' and add the following sentences: '... can be used as forcing parameters. **If the snow height (or snow water equivalent) and/or surface temperature are also specified in the input file, these are used as initial conditions. Otherwise, snow depth and surface temperature are assumed to be homogenous in space at the start of the simulation according to the specifications in the configuration file**.'

S: Done

R: **After the model description, the model is applied to a Tibetan glacier as an example. I appreciate that you show that the model is indeed producing reasonable results, but I would have liked a bit more evaluation, analyses and interpretation on how well it is doing, and why there are differences, compared to observations and to other models.**

A: In this contribution we focus on the model description, but of course evaluation and interpretation are very important as well. It is difficult to find good glaciological data to compare different model versions but we are going to add the comparison of ablation in specific periods between COSIPY output and ablation stake readings. Furthermore, we will include profile plots of snow layer properties using data from Hintereisferner in the European Alps as a second example. We will also consider using data of the ESM-SnowMIP intercomparison project as suggested by the other reviewer. However, we emphasize that COSIPY is a glacier mass balance model and not a snow model so that a specific difficulty could be the prescribed soil temperature in the ESM-SnowMIP project. Nevertheless, we may use some of the metrics for the evaluation of COSIPY. However, a model intercomparison clearly is beyond the scope of this contribution.

C: We will make the appropriate changes to the existing text in Chapter 5 and will introduce new paragraphs/sub-chapters on the new datasets and evaluations.

S: Done

<u>**Abstract**</u>

R: P1 Lines 1-7 are a very general introduction. Is that necessary in an abstract? I suggest to either remove it or make it much shorter. Formulations are also not clear. For example, 'key role' in what (line 1)? and where do 'these changes' (line 2) refer to?

A: The comments of the expert are reasonable. Since GMD is not a glaciological journal, we thought to create a broader context why distributed mass balance models are needed. But if this context is too broad we will shorten the first lines.

C: "**Glacier changes are a vivid example of how environmental systems react to a changing climate.** Distributed surface mass balance models which translate the meteorological conditions on glaciers into local melting rates **help to** attribute and detect glacier mass and volume responses to changes in the **climate drivers**. A well ...."

S: Done.

R: P1 L8: remove 'lean'. I have no idea what you mean by this.

A: The term "lean" is derived here from "lean concept". A lean design understands the requirements of the model user and focuses on continuously improving the handling of the model without unnecessarily expanding the model environment.

C: If this term is confusing we will remove it.

S: Done.

R: P1 L16: remove 'in'.

A/C: Will be done.

S: Done

<u>**Introduction**</u>

R: P2 L2: What do you mean with 'many scientific aspects'?

A: The chosen expression is probably unfortunate. What was meant was rather the perspectives on various scientific questions.

C: We will replace 'scientific aspects' with the term '**scientific issues**'.

S: Done

R: P2: Note also the work by Ostby et al. 2017 TC, and by van Pelt et al. (several studies) for Svalbard.

A: We are aware that there are very good and mentionable studies on this topic. We have tried to present a selection that covers the wide range of mass balance studies. If one or the other study is not listed, it is not intentional.

C: We will add the work of Ostby et al. 2017 and van Pelt to the reference list.

S: Done

R: P2 L30: The Hock and Holmgren 2005 JGI model is available on github.

A/C: We are aware that this model is available on github and we have mentioned this work in line 28.

S: Done

R: P2 L33: Remove 'lean'.

A/C: If this term is confusing we will remove it (see comment above).

S: Done.

R: P3 L6: Make much more clear what is new. I do not see it.

A/C: See response above in general comments.

S: Done

Model concept

R: **eq(3): The second term on the right hand side reads: $k_s \frac{\delta^2 T_s}{\delta z^2}$. Shouldn't this be $\frac{\delta}{\delta z}(k_s \frac{\delta \;\; T_s}{\delta z})$.? In your case you ignore the effect of the gradient of k with depth. Furthermore, what is the functional form you take for k_s? And why use it? Bartelt and Lehning already note that they think this is an inferior description of k_s.**

A: In general form this equation should indeed reads as $\frac{\delta}{\delta z}(k_s \frac{\delta \;\; T_s}{\delta z})$ but since in our model k_s does not depend on T, k_s is assumed to be a constant (average over the considered layers where the derivative is calculated - hence its a bulk conductivity). Thus the equation reduces to $k_s \frac{\delta^2 T_s}{\delta z^2}$. This simplifies the calculation and allows for solving the equation using a linear equation system. We agree that better results may be obtained when k_s depends on the spatial variable, e.g. $k_s(z)$. The equation becomes nonlinear and slightly more complicated. A gauge transformation could eliminate the spatial dependency and reduce the equation to $k_s \frac{\delta^2 T_s}{\delta z^2}$.Right now this is not implemented in the model, so that the given equation is correct. However, we agree that we should keep in mind that using a nonlinear heat equation would be an improvement of the model.

We are not exactly sure what the reviewer means with functional form and 'why use it'. The comment probably relates to the calculation of k_s. As given on p4L3, the volumetric thermal conductivity is calculated by the volumetric fractions of ice, water and air. The thermal conductivities for the constitutes are assumed to be constant (values are given in Appendix A).

Bartelt and Lehning indeed find the volumetric conductivity inferior to empirical or microstructural thermal conductivity models. The latter cannot be implemented as COSIPY does not model the microstructure of snow or ice. An option would be an empirical form of $k_s$depending on density and/or temperature. For sake of consistency with COSIMA, we will add the empirical form, $k_s = 0.021 + 2.5(\rho_s/1000)^2$, suggested by Anderson (1976). The user can then choose between the two forms.

C: We will point out in p4L3 that this is a bulk thermal conductivity, given by the average of the involved layers, and that a linear system of equations is used to solve the heat equation. We will also add the empirical thermal conductivity equation based on density (see above) with the note that the user can choose between these two options.

S: We have added: "Alongside the volume-specific heat capacity, COSIPY also offers the option of using empirical form k_s= 0.021 + 2.5(ρs/1000.0)^2".

R: **P3 L25: check the equation, for cs in combination with eq(3). I think there is a ρ_s to many in eq(3).**

A: Thank you for pointing this out.

C: We have removed the fractional densities from c_s.

S: Done

R: **P4 L7: Is your model indeed as deep that it reaches the base of the glacier? Most models only go 20 to 30 m deep. More is not really necessary for climatic surface mass balance studies.**

A: In fact, the domain does not always go to the base of the glacier and the statement is wrong. The user can determine the maximum depth of the computing domain. The default setting is 20 m.

C: We'll rewrite the sentence to '***At the bottom of the domain***, …'.

S: Done

R: **P4 L16: In my own experience, re-meshing, complete making of a new grid, is not necessary to do every time step, but can be made depended on melt and snow fall. This speeds up the model considerably when nothing is happening to the snowpack. Or do you also refer to re-meshing when only thickness of the layers changes a little, and thus also depth, due to densification?**

A: If the logarithmic approach is chosen, the remeshing is executed at each time step. This means that every change in layer thickness due to settling, densification, snowfall or melt triggers the remeshing algorithm. The logarithmic profile is defined by the thickness of the top layer and a stretching which is specified by the user. So far we have not thought about making the logarithmic remeshing conditional. But we will try this and hope for a speedup. Thanks for the helpful hint.

C: Within the scope of this COSIPY version we refrain from a conditional remeshing algorithm, but will test it in the next version. Therefore we see no reason to change the manuscript at this point.

S: Done

R: **P4 eq(5) Where does this equation come from? Coleou and Lesaffre, 1998, provides 1 equation for the full range of**

A: This is the same equation used by the study of Wever et al. (2014), which used exactly this formulation.

C: We will add this reference.

S: Done.

R: **P5 L1: Does the model include saturation of the snow? And if so, how is it described, and if not, please mention.**

A/C: Yes. But maybe we don't really understand the question. Each layer can retain water up to its retention capacity (see Eq. 5). Only if the capacity is exceeded, the excess water is transported to the next layer. This approach corresponds to the commonly used bucket approach. When the liquid water content reaches the retention capacity, the snow is saturated. How this is treated in the model is described in the text (p4L27): "In case the liquid water content of a layer exceeds its retention capacity … the excess water is drained into the subsequent layer (bucket approach)".

S: Done.

R: **P5 L2: What happens in the accumulation/firn area? When does runoff occur in that area?**

A: This is an exciting question. Up to now, a snow-ice threshold value can be defined by the user, which determines from which density on snow is referred to as ice. The threshold is usually set around 900 kg m^{-3}. Water percolates up to the first layer that is greater than or equal to this density and is then regarded as runoff. If there is no such layer, water percolates through the lower boundary of the domain and is then considered as runoff. We will clarify in the text how water percolation is treated in the accumulation/firn area.

C: "The liquid water is passed on until it reaches **either a layer of ice or** the surface of the glacier, where it is considered to be runoff. **For this purpose a threshold value was introduced which**

defines the transition from snow to ice. If no such layer exists, water is passed on until it reaches the lower limit of the domain and is then considered as runoff."

S: Done

R: **P5 L17: Especially with respect to solar radiation it is important to mention how you distribute the input forcing over the glaciers. Do you include a formulation to distinguish between direct and diffuse radiation, shading? Or does the user have to do that separately?**

A: We agree. As mentioned above, COSIPY does not provide any standard pre-processing routines and it is up to the user how to prepare or spatially interpolate the data. However, we do provide example scripts that illustrate and facilitate the data preparation workflow for the user. An example is given in the online documentation. The example shows how the input dataset can be generated from automatic weather station data and a given digital elevation model. The example scripts use simple lapse rates for temperature, precipitation and humidity for the interpolation. The wind speed is assumed to be constant over the domain. For the interpolation of the radiation the radiation model of Wohlfahrt et al (2016) is used (doi: 10.1016/j.agrformet.2016.05.0120). The current implementation does not distinguish between direct and diffuse radiation, but considers the total incoming solar radiation.

C: We will add a sentence in chapter 4 to clarify that the pre-processing of the data must be done by the user: The paragraph now reads as:

"*The model is driven by meteorological data that must be provided in a corresponding NetCDF file (see https://cosipy.readthedocs.io/en/latest/Ressources.html). Input parameters include atmospheric pressure, air temperature, cloud cover fraction, relative humidity, incoming shortwave radiation, total precipitation and wind velocity. Optional snowfall and incoming longwave radiation can be used as forcing parameters. In addition to meteorological parameters, COSIPY requires static information such as topographic parameters and a glacier mask.* ***Example workflows for creating and converting static and meteorological data into the required NetCDF input is included in the source code (*****https://cosipy.readthedocs.io/en/latest/Documentation.html#quick-tutorial)*****.*** *Besides the standard output variables, there is also the possibility to store vertical snow profile information (not recommended for distributed simulations). To reduce the amount of data, the users can specify which of the output variables will be stored*."

We will change the phrase 'Various tools are available ...' to '***Example workflows for creating and converting static and meteorological data into the required NetCDF input is included in the source code (https://cosipy.readthedocs.io/en/latest/Documentation.html#quick-tutorial***)'.

S: Done

R: **P6 L2: Also in case of longwave radiation, how do you distribute this over the glacier? Do you then always use eq (16)?**

A: There are two possibilities here. The user can specify the distributed longwave radiation in the input data or the longwave radiation is calculated using the Stephan-Boltzmann law and atmospheric emissivity (Eq. 15). Eq. 16 is included in the calculation of the atmospheric emissivity. As with all other input data, the longwave radiation must be distributed over the topography by the user.

C: See comment above.

S: Done

R: **P6 eq(17,18): I do not understand the term 1/Pr in this equation. In my opinion this factor should be included in how you calculate Ch and Ce, since not all methods that you present to calculate Ch and Ce should include this term.**

A: Thank you very much for this hint. This is actually an error in the formulation. We will remove 1/Pr from the Eq. (17) and Eq. (18) and also correct the Equations 25-27.

C: We will correct the Equations 17 and 18.

S: Done.

R: **P7 L1: How do you determine z0q and z0t, you only mention a factor. Do you indeed only apply a factor on z0m to obtain z0q or z0t or do you use a method such as described by Andreas, 1987, BLM?**

A: The two roughness lengths z0q and z0t are derived from z0v and are in fact one or two orders of magnitude smaller. Currently, the two roughness lengths are not parameterized separately as indicated in the text: "*The aerodynamic roughness length z0v is simply a function of time and increases linearly for snowpacks from fresh snow to firn (Mölg et al., 2012). For glaciers, z0v is set to a constant value. According to the renewal theory for turbulent flow, z0q and z0t are assumed to be one and two orders of magnitude smaller than z0v, respectively (Smeets and van den Broeke, 2008; Conway and Cullen, 2013)*".

C: We think these sentences adequately describe how the roughness lengths are derived and thus do not make any changes.

S: Done

R: **P8 L4: chang to 10 superscrip (-2). Confusing as it is now.**

A/C: Will be done.

S: Done.

R: P8 L11-15: What you describe here is a variation on what is described in Bartelt and Lehning. However, you refer here to a French report, which is hard to find. I prefer you to change the reference to something that is general available.

A: As far as we know, Bartelt and Lehning use a microstructure based viscosity formulation. Eq. 32 originates from Anderson (1976) but the parameter values used by default are those from Boone (2004).

One way around this reference would be to cite another paper that used the same parameterization, e.g. Essery et al (2013).

C: We will, therefore, also include the reference Essery et al. (2013) in this paragraph.

S: Done

Example

R: **P12 L10: Please make more clear that you start by running the model for a single location and only run it in distributed mode from line 30 onwards. It is often not clear whether you refer to a result for a single location or the whole glacier.**

A: Yes we agree that this has to be better pointed out in the text.

C: We will change P12 L12 to: *"**As a first example, we** use hourly data from May 2009 to June 2012 from an automatic weather station (AWS) on the Zhadhang Glacier (Huintjes et al.,2015) **fo force COSIPY as a point model for a single location**." and P12 L30 to: "**For a distributed glacier-wide run we drive COSIPY by ERA5 data instead of in-situ observations.** The glacier-wide cumulative surface mass balance for the decade 2009 to 2018 is presented in Figure 2. The computational domain consisted of 1837 grid cells with a spatial resolution of approximately 30 m (1 arcsecond) (see Fig. 2).*

S: Done

R: **P12 L14: Capital RH instead of rH.**

A/C: We will change the abbreviation for relative humidity to capital RH.

S: Done

R: **P12 L16: Where do you get the precipitation from?**

A: We got the accumulated precipitation from a sonic ranger.

C: We will change the sentence to: *"The relevant variables air pressure pzt , air temperature Tzt , relative humidity rH zt , incident short-wave radiation qG**, snowfall SF** and wind speed uzv were measured by the AWS."*

S: Done

R: **P12 L24: You first have to mention that you obtain the surface temperature from Lout observations, else this statement makes no sense. At this point I would like a bit more**

information on when the model is doing a good job, and when it struggles, and why. What are the limitations, how does this compare to other distributed mass and energy balance models?

A: Yes that is true. We will mention that the surface temperature is obtained from longwave radiation measurements. Beyond that we cannot compare in detail to other models without running these on the same dataset. However, a model intercomparison is beyond the scope of this manuscript.

C: We will change the sentence (P12 L21) to: *"Figure 1a and 1b show the glacier surface temperatures **determined from longwave radiation measurements** for two periods where in-situ measurements were available."*
As mentioned at the beginning of the document for the general comment on the Zhadang example, we will make the appropriate changes to the existing text in Chapter 5 and will introduce new paragraphs/sub-chapters on the new datasets and evaluations.

S: Done

R: **P12 L24: Where is the stake you refer to here located with respect to the weather station and your grid point?**

A: The stake is in the vicinity of the weather station. It is located within the same grid point of the simulation as the weather station. We will add more stake data to the evaluation of the results and point out much clearer where the stakes are located in the revised version.

C: According to the answer above, we will change the whole paragraph concerning the ablation stakes and compare only melt periods between model and observations.

S: Done

R: **P12 L26: Typo: modelleld should be modelled.**

A/C: Will be changed to modelled.

S: Done

R: **P12 L24-30: In this analyses I suggest that you distinguish between the time the glacier is snow covered and when the ice surface appears. The model should be well capable to reproduce the amount of ice melt, whereas surface changes in case of snow cover are much more difficult, since that also includes firn densification processes. Presenting ice melt separately also gives an indication of how well you reproduce the energy fluxes at the surface. Unless this is all snow covered period. But you have to make that clear. Figure 1: Make clear whether this is m ice/snow or m w.e. And indicate in the figure when ice is exposed (or not).**

A: In Figure 1, it is m ice/snow not m w.e. We will clarify this in the capiton and we agree that this will be a valuable improvement to the manuscript to better distinguish between snow melt and ice melt. As mentioned above, we will add the comparison between single melt periods which we can constrain using stake readings and compare the ablation phase between the model and the stake readings for those.

C: According to the answer above, we will make the appropriate changes to the existing text in Chapter 5 and will introduce new paragraphs/sub-chapters on the new datasets and evaluations.

S: Done

R: **P12 L33 - P13 L2: When you refer here to the distributed version of the model, do you compare the grid point of the weather station with the results when running the model only for the weather station location? If that is the case, why is there a difference in annual mass loss? What is done differently?**

A: Thank you for pointing this out. We will change this in the text because it does not make sense to compare the distributed glacier-wide mass balance to the single point simulation at the weather station.

C: We will compare here the glacier-wide run only with the literature values and not with the run for the location of the weather station. We will skip the comparison with the weather station location and change P12 L32 - P13 L3 to::

*"The **simulated mass balance** during this period was −**1.5** m w.e. **a^{-1}**. The results are in line with the analysis of Qu et al. (2014) who reported negative mass balances of −1.9, −2.0, −0.8 and −2.7 m w.e for the years 2009 to 2012."*

S: Done

R: **P13 L2: To what results do you refer here? Glacier wide? or Point location? Or the difference between them?**

A: In this case we refer to the glacier wide run. We deleted the sentence with the comparison to the weather station data. After that correction it is more obvious that from P12 L30 on we talk only about the distributed ERA5-forced run.

C: See proposed changes to reviewer comment above.

S: Done

Conclusion

R: **P15 L8: Remove 'of its kind'.**

A/C: Will be done.

S: Done

R: **P15 L12-17: Other models can do this as well, and moste of the topics mentioned have been done, at least for individual regions. What does this model add to that?**

A: There are a large number of mass balance models for snow and glaciers, but as mentioned at the beginning of this revision, there are significant differences with regard to implementation and model structure. COSIPY does not compete with existing models, but offers an accessible model structure with commonly used parameterization in glaciology, while micro-structural process required for detailed snow simulation are neglected (e.g. the micro-structure) as they are built into more sophisticated models such as SNOWPACK or CROCUS. The implementation in Python and the easy access for users makes the model attractive for glaciological applications which is also reflected in the increasing number of users.

C: We will summarize these advantages as well as the disadvantages in the conclusion.

S: Done

REVIEWER 2:

R: The manuscript by Sauter et al., describes the COSIPY v1.2 open-source coupled snowpack and ice surface energy and mass balance model. This model is designed to simulate the energy and mass balance of snow and ice covered surfaces, with applications for glacier mass balance simulations. The model builds on several decades of research in the field of snow cover and ice simulations. The main originality of this model is that it is implemented in Python. Given the scope and the content of the manuscript, it is fully appropriate for publication in Geoscientific Model Development. Overall, the manuscript reads well and I have not identified major flaws in the manuscript. Note, however, that I haven't checked one by one all the equations in Sections 2 and 3, which are based on classical concepts and frameworks for snow and ice energy and mass balance. I have several comments, which can rather be seen as suggestions, to the authors, and a series of minor comments.

Section 4 : While Sections 2 and 3 are in fact of limited added-value given that the equations are concepts are already outlined in a number of previous publications (it is fine to leave them in the manuscript, this is a useful reference for users of the model or its output, perhaps complemented by recent publications such as Essery et al., 2013, http://dx.doi.org/10.1016/j.advwatres.2012.07.013 and Lafaysse et al., 2017, http://dx.doi.org/10.5194/tc-11-1173-2017), I find section 4, addressing "Model architecture", quite short and it could be expanded to better address the novelty and added value of the model compared to previously existing models. For example, I think that it could be useful to provide more details regarding the Python libraries used for this model, their common dependencies, added-value, etc., and how the "modularity" of the model structure is implemented. This could be addressed not only by adding text, but also figures, providing an overview description of the model structure and the interlinkages between them.

A: We will complement the Sections with the work done by Essery et al. (2013) and Lafaysse et al. (2017) at the corresponding locations. Both are excellent references.

Originally we had planned another paragraph about the modular structure of the model. However, we decided to not integrate it, as we felt that this technical information would be better presented in the readthedocs online documentation (https://cosipy.readthedocs.io/en/latest). We will, however, add a paragraph about the code implementation, dependencies and structure of the model. Some information is already given in Section 7 ‘Code availability, documentation, and software requirements’, which is mandatory in GMD. Additionally, we will also discuss the added-value of COSIPY compared to previous glacier mass balance models. This has already been criticized by the first reviewer.

C: In Section 4, we will take up this issue again and highlight the special features. These include

- is completely written in Python, modular and object-oriented
- completely based on open-source libraries
- has a readthedocs documentation (which is still in the development phase)
- parameterizations can be easily extended or modified by the user
- NetCDF IO routines
- easy integration of Weather Research and Forecast (WRF) forcing
- adapted for distributed glacier mass balances simulations; it needs to be pointed out in this regard that COSIPY is not a snow cover model
- has a community platform (Slack) and code is actively maintained
- new git commits are automatically tested via travis and codecov
- each model version gets a DOI
- has a restart option for operational applications

S: Done

R: Section 5 : The section 5 provides an example of the model use for the Zhadang glacier, High Mountain Asia, with illustrations of model output (Figures 1 and 2) and model performance (Figure 3) for this case study. The results appear to be reasonable for a typical energy and mass balance model applied to a glacier setting. However, this does not correspond to a full model evaluation exercise, and I think this model description article would greatly benefit from a more robust evaluation. In this respect, I think the dataset used for the ESM-SnowMIP intercomparison could be particularly useful. All the relevant data have been made available in Ménard et al., 2019 (https://doi.org/10.5194/essd-11-865-2019), and paper such as Krinner et al., 2018 (https://doi.org/10.5194/gmd-11-5027-2018) can be used as inspiration for providing the evaluation indicators of snow models. Regardless of how this is handled, I consider useful for this model description article to provide some evaluation metrics relevant to the performance of the model described in this article.

A: Thank you for the suggestion. We agree it would be of great benefit to include a comparison with other models, however a full model intercomparison is beyond the scope of this manuscript.. We will use the data from Ménard et al., 2019 as forcing data to reproduce the metrics and compare COSIPY to other models similar to Krinner et al., 2018. It might be that we can not reproduce all metrics since COSIPY is a glacier energy and mass balance model and not a snowmodel, i.e., the soil heat flux is not parameterized in the same way, for example. Furthermore, we will extend the model evaluation for the Zhadang glacier with more ablation-stake data and present profile plots of the layer properties for the Hintereisferner in the European Alps. This exercise is also in response to comments of the other reviewer.

C: We will make the appropriate changes to the existing text in Chapter 5 and will introduce new paragraphs/sub-chapters on the new datasets and evaluations.

S: Done

R: Page 4, line 15 : how do the re-meshing algorithms compare to existing re-meshing algorithms used in other snow cover models ? I think in particular of Crocus (Vionnet et al., 2012, https://doi.org/10.5194/gmd-5-773-2012), there are other models with remeshing approaches. I think it would be good to position the approach taken here within other existing models.

A: Similar to CROCUS, COSIPY uses a set of criteria that determine when two layers are merged or splitted. As already indicated in the text the user can choose between two options:

(i) Logarithmic profile: This method is only suitable for simulations where the layering of the snowpack is not relevant (bulk). Layer thicknesses are calculated starting from the top layer, which always remains at a constant thickness, and gradually increases with depth by a constant stretching factor. Thus the layers close to the surface have a higher spatial resolution, which is advantageous for the computation of the energy and mass fluxes at the surface.

(ii) Adaptive profile: The adaptive algorithm runs in three consecutive steps: (1) adding/removing snow/ice at the surface, (2) adjusting the first layer, (3) updating internal layers.

> In the first step it is checked whether snow falls or melts away (note: internal layers can also melt). If snow falls on the glacier surface, it will only remain on the surface if it reaches a user-defined minimum snow thickness. If it falls on an existing snowpack, any snowfall that exceeds a user-defined minimum threshold is added to the snowpack. Melt is removed from the first layers and internal layers. After this step, layers can become very small and the thickness of the first layer no longer corresponds to the user-specified constant thickness. Therefore, it is necessary to remesh the layers.
>
> In the second step, the top layer is adjusted first. The top layer is remeshed so that this layer always has the user-defined layer thickness (default value is 0.01 m). The adaptation of the top layers together with internal melting processes can reduce the internal layers to a very low thickness. To avoid thin layers, the layers are merged or split in the last step (see next paragraph).
>
> In the last step, internal layers are splitted or merged. For each layer, a check is made to identify layers with a thickness of less than a defined minimum layer thickness. Such thin layers are merged with the layer below. Also if the differences in temperature and density of two subsequent layers are less than a user defined threshold (similarity criteria), they will be merged. How often a merging/splitting can take place per time step is also defined by the user (correction steps). Unlike CROCUS, internal remeshing always starts from the surface, i.e. the uppermost layers are adapted first. Depending on how many correction steps are set by the user, it can happen that only the uppermost layers are remeshed.

C: We will extend the description of the meshing algorithms (beginning from page 4 line 15) and describe them in more detail as outlined above.

S: We have divided the chapter 'Model Concepts' into two subchapters: (1) Fundamental Equations and (2) Discretization and computational mesh. In the latter, the re-meshing is now described in detail.

R: Page 4, line 24 : "useful feature" : would it be possible to elaborate on what is meant by "useful feature" ? What metric was used to address the "usefulness" ?

A: The term 'useful feature' should indicate that the adaptive algorithm is reasonable when one is interested in the stratification of the snowpack. In contrast to the logarithmic profile, one obtains well resolved layers. This can be important for some glaciological issues, but it is computationally more expensive than the logarithmic algorithm.

C: We will rewrite the sentence "*The adaptive re-meshing proves to be a useful feature, but slightly increases both the computing time and the data volume*" to "***Unlike the logarithmic approach, adaptive re-meshing resolves individual layers but slightly increases both computing time and data volume.***"

S: Done

R: Page 7, line 3 : I suggest to use the LaTeX symbol \varepsilon instead of \epsi, this seems to better match the graphical design of the "epsilon" symbol, when it refers to the emissivity.

A/C: Will be done.

S: Done

R: Page 9, line 3, I suggest replacing "Von" by "von" for the name of "von Karman" (ideally with "accents" on the "a"s).

A/C: Will be done.

S: Done

R: Page 10, line 7 : I don't think it is adequate to refer to "snow grain settling", but "Snow settling" would be less ambiguous and more accurate.

A/C: We agree and will change it to "snow settling".

S: Done

R: Page 11, line 20 : I think more details should be given on what is referred to here as "dynamic mesh" ?

A: The wording is probably a bit confusing. The term should refer to the computational mesh which can be (dynamically) adapted by the re-meshing algorithms.

C: To clarify this misunderstanding we will remove the term 'dynamic'.

S: Done

R2: P12 L9: More explanations could be given to better explain the content of the parenthesis "(not recommended for distributed simulations)"

A: The user can specify in a file which data should be stored. In addition to the atmospheric variables, COSIPY calculates the state of the snow/ice layers. Since the number of vertical layers and grid cells can be very high, it is recommended to store only those variables that are necessary for later evaluation. We recommend dropping the states of the layers in distributed simulations to save memory space.

C: We will change the sentence to: "*Besides the standard output variables there is also the possibility to store vertical snow profile information**, although to save memory we can only recommend this for single point simulations.***"

S: Done

R2: P12 L31 : I think more explanations are needed for "driven by ERA-5", in particular whether downscaling was applied, and if yes, how.

A: We extracted the needed input data from the nearest ERA5 grid point and applied downscaling methods to the variables. We will further clarify this in the revised version of the manuscript.

C: We will add a table to the manuscript with the applied downscaling approaches and change the sentence to: *"The model was driven by ERA5 data instead of in-situ observations.*
The ERA5 data were downscaled to the site using straightforward approaches. Temperature, T_{zr}, and humidity, RH_{zt}, were corrected to the altitude of the grid cell using empirical lapse rates. For pressure, p_{zt}, the barometric formula was used. The radiation model of Wohlfahrt et al (2016) was used for the incoming shortwave radiation to account for effects of shadowing, slope and aspect. Total precipitation RRR, N and U_{zv} were used directly from the closest ERA5 grid point."

S: Done

R2: P13 Figure 1: I suggest replacing "modelled" by "simulated" in the legend and captions.

A/C: We agree and will change it accordingly in all legends and captions.

S: Done

R2: Page 14, Figure 3: Would it be possible to provide a definition for the term "Speedup"? I think this would be a useful clarification. If possible, it would be useful to provide a comparison of this metric with other existing models, in order to address to what extent the scalability of this Python-based model is comparable to implementations using other programming language.

A: Thank you for the important comment. The speedup is the ratio between the single-core execution time and the execution time of the corresponding multiple-core simulation. We wanted to point this out with the sentence: "..., i.e. the ratio of the original execution time with the execution time of the corresponding node test." In the present case, it is difficult to compare this value with other models because we would have to run the other models to the same test case. The speedup in the present case should rather show the performance gain when using multiple cores in contrast to a single core computer setup.

C: We will change the caption of Figure 3 to: *"Speedup **(execution time of single-core simulation divided by execution time of the corresponding multiple-core simulation)** for computing a 10-year distributed COSIPY run on Zhadang glacier with 206 grid points."* and the respective sentence to: *"..., i.e. the ratio of the original execution time **(single core)** with the execution time of the corresponding test **(multiple cores)**."*

S: Done

R: Page 15, line 15 : I think it would be appropriate to also refer to multiphysics modelling, and it would be good to know to what extend COSIPY can be used for such applications (see e.g. Pritchard et al., 2020, https://doi.org/10.5194/tc-14-1225-2020).]

A: Multiphysical modeling is a very exciting topic and we are convinced that it will become even more important in the future. COSIPY is a modeling platform designed to test and apply different parameterizations. In principle it is already possible to generate ensemble simulations with different physical parameterizations and solvers, but COSIPY is not yet an ensemble multiphysics modeling environment. As a vision for the future it is conceivable to extend COSIPY for automatic ensemble simulations. Various uncertainties could be included - multiphysics modeling, perturbed input data or parameter uncertainty. So far, we have not yet thought about how an ensemble can be realized in a single simulation. Until we have a feasible idea, we have no choice but to run COSIPY with different combinations of physical parameterizations or input uncertainty and evaluate the statistics afterwards.

C: At this point we will, as an addition to the existing manuscript, only provide an outlook on what might be possible with COSIPY in the future, e.g. ensemble simulations.

S: Done

COSIPY v1~~.2~~ .3 - An open-source coupled snowpack and ice surface energy and mass balance model

Tobias Sauter[1], Anselm Arndt[2], and Christoph Schneider[2]

[1] Department of Geography, Friedrich-Alexander-Universität Erlangen-Nürnberg, Wetterkreuz 15, 91058 Erlangen, Germany

[2] Geography Department, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany

Correspondence: Tobias Sauter (tobias.sauter@fau.de)

Abstract. ~~Glacial changes play a key role both from a socio-economical and political, and scientific point of view. The identification and the understanding of the nature of these changes still poses fundamental challenges for climate, glacier and water research. Many studies aim to identify the climatic drivers behind the observed glacial changes using distributed surface mass and energy balance models.~~ Glacier changes are a vivid example of how environmental systems react to a changing climate. Distributed surface mass balance models, which translate the meteorological conditions on glaciers into local melting rates ~~, thus offer the possibility~~ help to attribute and detect glacier mass and volume responses to changes in the ~~climatic forcings~~climate drivers. A well calibrated model is a suitable test-bed for sensitivity, detection and attribution analyses for many scientific applications and often serves as a tool for quantifying the inherent uncertainties. Here we present the open-source coupled snowpack and ice surface energy and mass balance model in Python COSIPY, which provides a ~~lean,~~ flexible and user-friendly framework for modelling distributed snow and glacier mass changes. The model has a modular structure so that the exchange of routines or parameterizations of physical processes is possible with little effort for the user. The framework consists of a computational kernel, which forms the runtime environment and takes care of the initialization, the input-output routines, the parallelization as well as the grid and data structures. This structure offers maximum flexibility without having to worry about the internal numerical flow. The adaptive sub-surface scheme allows an efficient and fast calculation of the otherwise computationally demanding fundamental equations. The surface energy-balance scheme uses established standard parameterizations for radiation as well as for the energy exchange between atmosphere and surface. The schemes are coupled by solving both surface energy balance and subsurface fluxes iteratively ~~in~~ such that consistent surface skin temperature is returned at the interface. COSIPY uses a one-dimensional approach limited to the vertical fluxes of energy and matter but neglects any lateral processes. Accordingly, the model can be easily set up in parallel computational environments for calculating both energy balance and climatic surface mass balance of glacier surfaces based on flexible horizontal grids and with varying temporal resolution. The model is made available on a freely accessible site and can be used for non-profit purposes. Scientists are encouraged to actively participate in the extension and improvement of the model code.

1 Introduction

Glacier variations are of great interest and relevance in many scientific ~~aspects~~ issues and application such as climate sciences, water resources management and tourism. In order to identify the climatic drivers for past, current and future changes, process understanding, observations and models of glacier mass change need to be combined appropriately. Schemes that relate the surface mass balance of snow and ice bodies to meteorological forcing data have been set up and applied since many decades (e.g. Anderson, 1968; Kraus, 1975; Anderson, 1976; Kuhn, 1979; Male and Granger, 1981; Kuhn, 1987; Siemer, 1988; Morris, 1989, 1991; Munro, 1991). Studies have shown that the synthesis of these information provides a consistent understanding of the relevant mass and energy fluxes at the glacier-atmosphere interface, which in turn provides the necessary physical foundations to translate micro-meteorological conditions on glaciers into local melt rates (e.g. Sauter and Galos, 2016; Wagnon et al., 1999; Oerlemans, 2001; Mölg and Hardy, 2004; Obleitner and Lehning, 2004; Van Den Broeke et al., 2006; Reijmer and Hock, 2008; Mölg et al., 2008; Nicholson et al., 2013).

Distributed mass balance models combine the local melt information to a glacier-wide surface mass change information and thus offer the possibility to attribute and detect glacier mass and volume responses to changes in the climatic forcings (e.g. Klok and Oerlemans, 2002; Hock and Holmgren, 2005; Mölg et al., 2009; Sicart et al., 2011; Cogley et al., 2011). Although the accumulation and redistribution of snow are still deficient (e.g. Sauter et al., 2013), when coupled with atmospheric models such models have the potential to simulate present and future glacier evolution or to serve as a useful tool for monitoring climatic glacier mass change (Machguth et al., 2006). A well calibrated model is a suitable platform for sensitivity, detection and attribution analyses as well as a tool for the quantification of inherent uncertainties ~~(e.g. Mölg et al., 2014; Maussion et al., 2015; Rye et al., 2012; Mölg et al., 2012; Sauter and Obleitner, 2015; Galos et al., 2017)~~ (e.g. Mölg et al., 2014; Maussion et al., 2015; Rye et al., 2012; Mölg et al., 2012; Sauter and Obleitner, 2015; Galos et al., 2017; van Pelt
.

Over recent decades, several distributed mass balance models of varying complexity have been developed and successfully applied to different glacier systems and climate regimes. The models range from simple degree-day models (e.g. Radić and Hock, 2006; Schuler et al., 2005) to intermediate models (e.g. Machguth et al., 2009) and complex snow cover and glacier resolving physical models ~~(e.g. Bartelt and Lehning, 2002; Vionnet et al., 2012; Hock and Holmgren, 2005; Klok and Oerlemans, 2002; Sica~~ (e.g. Bartelt and Lehning, 2002; Vionnet et al., 2012; Hock and Holmgren, 2005; Klok and Oerlemans, 2002; Sicart et al., 2011; Weidema
. The latter model class is usually based on the same fundamental physical principles but differ in the parameterisation schemes and implementation techniques. Different research groups have their own in-house solutions which are often extended and modified for specific scientific questions and studies. The fact that often several sub-versions of the same model exist, with some of them being not freely available, makes it difficult for users to having access to up-to-date software. Ideally, a platform should (i) be continuously maintained, (ii) provide newly developed parameterisations, (iii) compile different model subversions developed for specific research needs, (iv) be easily extensible and (v) be well documented and readable.

Here we present an open-source coupled snowpack and ice surface energy and mass balance model in Python (COSIPY) designed to meet these requirements. The structure is based on the predecessor model COSIMA (COupled Snowpack and Ice surface energy and MAss balance model, Huintjes et al., 2015b). COSIPY provides a lean, flexible and user-friendly framework for modelling distributed snow and glacier mass changes. The framework consists of a computational core that forms the run-time environment and handles initialization, input-output (IO) routines, parallelization, and the grid and data structures. In most cases, the runtime environment does not require any changes by the user. To increase the user-friendliness, additional features are available, such as a restart option for operational applications and automatic comparison between simulation and ablation stakes. These features will be further refined in the future. Physical processes and parameterisations are handled separately by modules. The modules can be easily modified or extended to meet the needs of the end user. This structure provides maximum flexibility without worrying about internal numerical issues. The model is completely based on open-source libraries and is provided on a freely accessible git repository (https://github.com/cryotools/cosipy) ~~and can be used~~ for non-profit purpose. Scientists can actively participate in extending and improving the model code. Changes to the code are automatically tested with Travis CI (www.travis-ci.org) when uploaded to the repository. It is planned to publish updates in regular intervals. To make working with COSIPY easier, a community platform (https://cosipy.slack.com) has been set up in addition to a detailed readthedocs documentation (https://cosipy.readthedocs.io/en/latest), allowing users and developers to exchange experiences, report bugs and communicate needs.

In this work, we describe the physical basics, parameterisations and outline the numerical implementation of the model version COSIPY v1~~.2~~.3 (https://doi.org/10.5281/zenodo.3613921). Section 2 gives an overview of the model concept, followed by the description of the modules (Section 3). The model architecture and the input/output are explained in Section 4). Section ~~?? shows an application~~ 5 shows different applications of the model. The last section (Section 7) documents the code availability and software requirements.

2 Model concept

2.1 Fundamental equations

COSIPY is a multi-layered process resolving energy and mass balance model for the simulation of past, current and future glacier changes. The model is based on the concept of energy and mass conservation. The snow/ice layers are described by the volumetric fraction of ice θ_i, liquid water content θ_w and air porosity θ_a. Continuity constraints require that

$$\theta_i + \theta_w + \theta_a = 1. \tag{1}$$

The inherent properties, such as snow density ρ_s or specific heat of snow c_s, follow from the volumetrically weighted properties of the constitutes. For example, snow density is related by

$$\rho_s = \theta_i \cdot \rho_i + \theta_w \cdot \rho_w + \theta_a \cdot \rho_a, \tag{2}$$

where ρ_i is the ice density, ρ_w the water density, and ρ_a the air density (Bartelt and Lehning, 2002). Exchange processes at the surface, the energy release and consumption through phase changes, control the vertical temperature distribution within the snow and ice layers. The energy balance also includes incoming shortwave radiation absorption and the sublimation or deposition of water vapour. Assuming the vertical temperature profile is given by $T_s(z,t)$, where z is the depth, the energy conservation can be represented by

$$\rho_s(z,t)c_s(\theta,z,t)\frac{\partial T_s(z,t)}{\partial t} - k_s(\theta,z,t)\frac{\partial^2 T_s(z,t)}{\partial z^2} = Q_p(z,t) + Q_r(z,t) \tag{3}$$

where ~~$c_s = \rho_i c_i \theta_i + \rho_w c_w \theta_w + \rho_a c_p \theta_a$ and $k_s = k_i \theta_i + k_w \theta_w + k_a \theta_a$~~ $c_s = c_i\,\theta_i + c_w\,\theta_w + c_p\,\theta_a$ and $k_s = k_i\,\theta_i + k_w \theta_w + k_a \theta_a$ are the volume-specific bulk heat capacity and bulk thermal conductivity of the snow cover (Bartelt and Lehning, 2002). Alongside the volume-specific heat capacity, COSIPY also offers the option of using empirical form $k_s = 0.021 + 2.5(\rho_s/1000.0)^2$ (Huintjes et al., 2015a). The first term on the right-hand side (Q_p) is the volumetric energy sink or source by melting and melt-water refreezing. The second term (Q_r) is the volumetric energy surplus by the absorption of shortwave radiation (see Eq. 13).

The exchange processes at the snow/ice-atmosphere interface control the surface temperature $T_s(z=0,t)$ at an infinitesimal skin layer. From the energy conservation follows

$$k_s(\theta, z=0,t)\frac{\partial T_s(z=0,t)}{\partial z} = q_{sw} + q_{lw} + q_{sh} + q_{lh} + q_{rr}, \tag{4}$$

where q_{sw} is the net-shortwave radiation energy, q_{lw} is the net-longwave radiation energy, q_{sh} is the sensible heat flux, q_{lh} is the latent heat flux, and q_{rr} is the heat flux from rain. To solve Eq. (4) for $T_s(z=0,t)$, the fluxes on the right-hand side must be parameterized (see Section 3). The parameterization results in a nonlinear equation which is solved iteratively. The left side of Eq. (4) provides the upper Neumann boundary condition (prescribed gradient) for Eq. (3). At the ~~base of the glacier~~bottom of the domain, the temperature must be specified (Dirichlet boundary condition) by the user. The melting rates in the snow cover and glacier ice are derived diagnostically from the energy conservation by ensuring that the temperature does not exceed the melting point temperature T_m.

~~To solve the underlying differential equations, the computing domain must be discretized. Since extreme gradients in temperature, density and liquid water content can develop in the snowpack, uses a dynamic, non-equidistant mesh. The mesh consists of so-called nodes that store the properties of the layers (e.g. temperature, density, and liquid water content), and is continuously adjusted during run-time by a re-meshing algorithm, i.e. the number and height of the individual layers vary with time. Currently, two algorithms are implemented: (i) A logarithmic approach, where the layer thicknesses gradually increase with depth by a constant stretching factor. Re-meshing is performed at each time step. This is a fast method, but does not resolve sharp layering transitions, as these are smoothed by the algorithm. This approach is only recommended if a detailed resolution of the snow and ice cover is not required. (ii) An adaptive algorithm that assembles layers according to user-defined criteria. It uses density and temperature thresholds to determine when two successive layers are considered similar and when~~

~~they are not. When both criteria are met, these layers are merged. The liquid water content and the heights of the two layers are added and the new density is calculated from the volumetrically weighted densities of the two layers. To ensure energy conservation, the total energy is determined from the internal energies and converted into the new temperature. The adaptive re-meshing proves to be useful feature, but slightly increases both the computing time and the data volume.~~

Eq. (4) is solved using a Limited Memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm (Quasi-Netwon method) for bound constrained minimisation (Fletcher, 2000). Eq. (3) is then integrated with an implicit second-order central difference scheme (Ferziger and Perić, 2002). The heat sources can warm the snowpack and lead to internal melt processes. In case the liquid water content of a layer exceeds its retention capacity ~~(Coléou and Lesaffre, 1998)~~(Coléou and Lesaffre, 1998; Wever et al., 2014) ,

$$\theta_e = \begin{cases} 0.0264 + 0.0099\,\dfrac{(1-\theta_i)}{\theta_i}, & \text{if } \theta_i \leq 0.23 \\ 0.08 - 0.1023\,(\theta_i - 0.03), & \text{if } 0.23 < \theta_i \leq 0.812 \\ 0, & \text{if } \theta_i > 0.812 \end{cases}, \tag{5}$$

the excess water is drained into the subsequent layer (bucket approach). The liquid water is passed on until it reaches either a layer of ice or the glacier surface where it is considered to be runoff. For this purpose a threshold value was introduced which defines the transition from snow to ice. If no such layer exists, water is passed on until it reaches the lower limit of the domain and is then considered as runoff. Meltwater refreezing and subsurface melting during percolation change the volumetric ice and water contents. Subsurface melt occurs when energy fluxes, e.g. penetrating shortwave radiation, warms the layer to physically inconsistent temperatures of $T_s > T_m$. Since physical constraints require that $T_s \leq T_m$, the energy surplus is used to melt the ice matrix. Melt takes place when $T_s > T_m$ and the liquid water content increases by

$$\Delta\theta_w(z,t) = \frac{c_i(z,t)\theta_i(z,t)\rho_i(z,t)(T_s(z,t) - T_m)}{L_f\rho_w}, \tag{6}$$

where $L_f = 3.34 \times 10^5 JKg^{-1}$ is the latent heat of fusion (Bartelt and Lehning, 2002). Mass conservation requires that the mass gain of liquid water content equals the mass loss of the volumetric ice content, so that

$$\Delta\theta_i(z,t) = \frac{\rho_w \Delta\theta_w(z,t)}{\rho_i}. \tag{7}$$

The latent energy needed by the phase change is

$$Q_p(z,t) = L_f \Delta\theta_i(z,t)\rho_i, \tag{8}$$

which is an heat sink because $\Delta\theta_i(z,t)$ is positive at melting. The energy used for melting ensures that $T_s(z,t)$ does not rise above T_m. In case $\theta_w > 0$ and $T_s < T_m$, refreezing can take place. Changes in volumetric fractions and the release of latent energy due to phase changes are treated equally. As the temperature difference must be negative due to the given constraints, it follows from Eq. (6), Eq. (7), and Eq. (8) that Q_p becomes positive and latent heat release warms the layer.

Many of the quantities and fluxes in Eq. (3) and Eq. (4) are not measured directly and have to be derived via corresponding parameterizations. The next section describes the parameterizations implemented in COSIPY v1~~.2~~.3.

2.2 Discretization and computational mesh

To solve the underlying differential equations, the computing domain must be discretized. Since extreme gradients in temperature, density and liquid water content can develop in the snowpack, COSIPY uses a dynamic, non-equidistant mesh. The mesh consists of so-called nodes that store the properties of the layers (e.g. temperature, density, and liquid water content), and is continuously adjusted during run-time by a re-meshing algorithm, i.e. the number and height of the individual layers vary with time. Currently, two algorithms are implemented: (i) A logarithmic approach, where the layer thicknesses gradually increase with depth by a constant stretching factor. Thus, layers close to the surface have a higher spatial resolution, which is advantageous for the computation of the energy and mass fluxes at the surface. Re-meshing is performed at each time step. This is a fast method, but does not resolve sharp layering transitions, as these are smoothed by the algorithm. This approach is only recommended if a detailed resolution of the snow and ice cover is not required. (ii) An adaptive algorithm that assembles layers according to user-defined criteria. It uses density and temperature thresholds to determine when two successive layers are considered similar and when they are not. When both criteria are met, these layers are merged. Basically, the adaptive algorithm runs in three consecutive steps: (1) adding/removing snow/ice at the surface, (2) adjusting the first layer, (3) updating internal layers.

(1) In the first step it is checked whether snow falls or melts away (note: internal layers can also melt). If snow falls on the glacier surface, it will only remain on the surface if it reaches a user-defined minimum snow thickness. Melt is removed from the first layer and all internal layers. After this step, layers can become very small and the thickness of the first layer no longer corresponds to the user-specified constant thickness. Therefore, it is necessary to re-mesh the layers.

(2) In the second step, the top layer is adjusted first. The top layer is re-meshed so that this layer always has the user-defined layer thickness (default value is 0.01 m). The adaptation of the top layers together with internal melting processes can reduce the internal layers to a very low thickness. To avoid thin layers, the layers are merged or split in the third step.

(3) In the third step, internal layers are splitted or merged. For each layer, a check is made to identify layers with a thickness of less than a defined minimum layer thickness. Such thin layers are merged with the layer below. Also if the differences in temperature and density of two subsequent layers are less than a user defined threshold (similarity criteria), they will be merged. How often a merging/splitting can take place per time step is also defined by the user (correction steps). Unlike CROCUS (Vionnet et al., 2012), internal re-meshing always starts from the surface, i.e. the uppermost layers are adapted first. Depending on how many correction steps are set by the user, it can happen that only the uppermost layers are re-meshed.

When two layers are merged, the liquid water content and the heights of the two layers are added and the new density is calculated from the volumetrically weighted densities of the two layers. To ensure energy conservation, the total energy is determined from the internal energies and converted into the new temperature. Unlike the logarithmic approach, adaptive re-meshing resolves individual layers but slightly increases both computing time and data volume.

3 Model physics and modules

3.1 Snowfall and precipitation

When snowfall is given, it is assumed that the data represents the effective accumulation since snowdrift and snow particle sublimation are not explicitly treated in the model. Otherwise, snowfall is derived from the precipitation data using a logistic transfer function. The proportion of solid precipitation smoothly scales between 100 % (0 °C) and 0 % (2 °C), as suggested by Hantel et al. (2000). The fresh snow density for the conversion into snow depth is a function of air temperature and wind velocity

$$\rho_s(z=0,t) \;=\; \max\left[a_f + b_f(T_{z_t} - 273.16) + c_f {u_{z_v}}^{1/2}, \rho_{min}\right], \tag{9}$$

with the empirical parameters $a_f = 109\,\mathrm{kg\,m^{-3}}$, $b_f = 6\,\mathrm{kg\,m^{-3}\,K^{-1}}$, $c_f = 26\,\mathrm{kg\,m^{-7/2}\,s^{1/2}}$, and $\rho_{min} = 50\,\mathrm{kg\,m^{-3}}$ (Vionnet et al., 2012). In both cases fresh snow is only added when the height exceeds a certain user-defined threshold.

3.2 Albedo

The approach suggested by Oerlemans and Knap (1998) parametrizes the evolution of the broadband albedo. The decay of the snow albedo at a specific day depends on the snow age at the surface and is given by

$$\alpha_{snow} \;=\; \alpha_f \;+\; (\alpha_s \;-\; \alpha_f)\;\exp\left(\frac{s}{\tau^*}\right), \tag{10}$$

where α_s is the fresh snow albedo and α_f the firn albedo. The albedo time scale τ^* specifies how fast the snow albedo drops from fresh snow to firn. The number of days after the last snowfall is given by parameter s. Besides the temporal change, the overall snowpack thickness impacts the albedo. If the thickness of the snowpack d is thin, the albedo must tend towards the albedo of ice α_i. If one introduces a characteristic snow depth scale d^* (e-folding) the full albedo can be written as

$$\alpha \;=\; \alpha_{snow} \;+\; (\alpha_i \;-\; \alpha_{snow})\;\exp\left(\frac{-d}{d^*}\right). \tag{11}$$

The model resets the albedo to fresh snow, if the snow accumulation exceeds a certain threshold (default value is 0.01 m) within one time step. This approach neglects sudden short-term jumps in albedo, which can occur when thin fresh snow layers quickly melt away. To account for this effect, the age of the underlying snow is also tracked. If the fresh snow layer melts faster than τ^* , the age of the snow cover is reset to the value of the underlying snow (Gurgiser et al., 2013).

3.3 Radiation fluxes

The net-shortwave radiation in the energy conservation equation Eq. (3) is defined as

$$q_{sw} = (1-\alpha)\cdot q_G, \tag{12}$$

where q_G is the incoming shortwave radiation, and α the snow/ice albedo. A proportion of the net shortwave radiation q_{sw} can penetrate into the uppermost centimetres of the snow or ice (Bintanja and Van Den Broeke, 1995). The resulting absorbed radiation at depth z is calculated with

$$Q_r(z,t) = \lambda_r\, q_{sw}\, \exp(-z\beta), \tag{13}$$

where λ_r is the fraction of absorbed radiation (0.8 for ice; 0.9 for snow), and β the extinction coefficient (2.5 for ice; 17.1 for snow). Physical constraints require that $T_s \leq T_m$ so that the energy surplus is used to melt the ice matrix (see Section 2).

In case the incoming longwave radiation $q_{lw_{in}}$ is observed, the net-longwave radiation is obtained by

$$q_{lw} = q_{lw_{in}} - \varepsilon_s \sigma {T_0}^4, \tag{14}$$

where ε_s is the surface emissivity which is set to a constant close ~~to~~ or equal to 1. In the absence of $q_{lw_{in}}$, the flux is parametrized by means of air temperature T_{z_t} and atmospheric emissivity,

$$\varepsilon_a = \varepsilon_{cs}(1-N^2) + \varepsilon_{cl} N^2, \tag{15}$$

using the Stefan-Boltzmann law. Here, N is the cloud cover fraction, ε_{cl} the emissivity of clouds which is set to 0.984 (Klok and Oerlemans, 2002), and ε_{cs} the clear sky emissivity. The latter is given by

$$\varepsilon_{cs} = 0.23 + 0.433\, (e_{z_t}/T_{z_t})^{1/8}, \tag{16}$$

where e_{z_t} is the water vapor pressure (Klok and Oerlemans, 2002).

3.4 Turbulent fluxes

The turbulent fluxes, q_{sh} and q_{lh}, in Eq. (4) are parametrized based on the flux-gradient similarity which assumes that the fluxes are proportional to the vertical gradient of state parameters. However, since meteorological parameters are only considered from one height in the model a bulk approach is used whereby the mean property between the measurement height and the surface is considered (e.g. Foken, 2008; Stull, 1988). Assuming that fluxes in the Prandtl layer are constant, dimensionless transport coefficients C_H (Stanton number) and C_E (Dalton number) can be introduced by vertically integrating the turbulent diffusion coefficients (Foken, 2008; Stull, 1988) so that the turbulent vertical flux densities can be written as

$$q_{sh} \;=\; \rho_a \, c_p \, C_H \, u_{z_v} \, (T_{z_t} - T_0) \tag{17}$$

$$q_{lh} \;=\; \rho_a \, L_v \, C_E \, u_{z_v} \, (q_{z_q} - q_0), \tag{18}$$

where ρ_a is the air density (derived from the ideal gas law), c_p is the specific heat of air for constant pressure, L_v is the latent heat of vaporisation which is replaced by the latent heat of sublimation L_s when $T_0 < T_m$, ~~(0.8) is the turbulent Prandtl number,~~ u_{z_v} is the wind velocity at height z_t, T_{z_t} and q_{z_q} are the temperature and mixing ratio at height z_t (assuming $z_t = z_q$), respectively, and q_0 is the mixing ratio at the surface where it is assumed that the infinite skin layer is saturated. Unlike the turbulent diffusion coefficients, the bulk coefficients are independent of the wind speed and only depend on the stability of the atmospheric stratification and the roughness of the surface. The aerodynamic roughness length z_{0_v} is simply a function of time and increases linearly for snowpacks from fresh snow to firn (Mölg et al., 2012). For glaciers, z_{0_v} is set to a constant value. According to the renewal theory for turbulent flow, z_{0_q} and z_{0_t} are assumed to be one and two orders of magnitude smaller than z_{0_v}, respectively (Smeets and van den Broeke, 2008; Conway and Cullen, 2013).

COSIPY provides two options to correct the flux-profile relationship for non-neutral stratified surface layers, by adding a stability correction using the (1) bulk Richardson-Number, and (2) Monin-Obukhov similarity theory (e.g. Conway and Cullen, 2013; Radić et al., 2017; Stull, 1988; Foken, 2008; Munro, 1989). Using the bulk Richardson number the dimensionless transport coefficients can be written in the form

$$C_H = \frac{\kappa^2}{\ln\left(\dfrac{z}{z_{0_v}}\right)\ln\left(\dfrac{z}{z_{0_t}}\right)}\Psi_{Ri}(Ri_b) \tag{19}$$

$$C_E = \frac{\kappa^2}{\ln\left(\dfrac{z}{z_{0_v}}\right)\ln\left(\dfrac{z}{z_{0_q}}\right)}\Psi_{Ri}(Ri_b), \tag{20}$$

whereas the stability function

$$\Psi_{Ri}(Ri_b) = \begin{cases} 1, & \text{if } \; Ri_b \;<\; 0.01 \\ (1 \;-\; 5\;Ri_b)^2, & \text{if } \; 0.01 \;\leq\; Ri_b \;\leq\; 0.2 \\ 0, & \text{if } \; Ri_b \;>\; 0.2 \end{cases}, \tag{21}$$

accounts for reduction of the vertical fluxes by thermal stratification and is a function of the Richardson number. The Richardson number

$$Ri_b = \frac{g}{T_{z_t}} \cdot \frac{(T_{z_t} - T_0)(z_t - z_{0_t})}{(u_{z_v})^2}, \tag{22}$$

follows from the turbulent kinetic energy equation and relates the generation of turbulence by shear and damping by buoyancy (Stull, 1988). In the stable case ($0.01 \leq Ri_b \leq 0.2$), the function describes the transition from turbulent flow to a quasi-laminar non-turbulent flow, and hence, reduces the vertical fluxes. Once Ri_b exceeds the critical value $Ri_b = 0.2$, turbulence eventually extinguishes, and the vertical exchange is suppressed.

According to the Monin-Obukhov similarity theory, atmospheric stratification can be characterised by the dimensionless parameter

$$\zeta = z/L. \tag{23}$$

where

$$L = \frac{{u_*}^3}{\kappa \frac{g}{T_{z_t}} \frac{q_{sh}}{\rho_a \cdot c_p}} \tag{24}$$

is the so-called Obukhov length with u_* is the friction velocity and κ (0.41) the von Kármán constant (Stull, 1988; Foken, 2008). The length scale relates dynamic, thermal and buoyancy processes and is proportional to the height of the dynamic sublayer. The bulk aerodynamic coefficients for ~~drag, heat and moisture~~ momentum C_D, heat C_H and moisture C_E for non-neutral conditions

$$C_D = \frac{\kappa^2}{\left[\ln\left(\frac{z}{z_{0_v}}\right) - \Psi_m(\zeta) - \Psi_m\left(\frac{z_{0_v}}{L}\right)\right]^2} \tag{25}$$

$$C_H = \frac{\kappa {C_D}^{1/2}}{\left[\ln\left(\frac{z}{z_{0_t}}\right) - \Psi_t(\zeta) - \Psi_t\left(\frac{z_{0_t}}{L}\right)\right]} \tag{26}$$

$$C_E = \frac{\kappa {C_D}^{1/2}}{\left[\ln\left(\frac{z}{z_{0_q}}\right) - \Psi_q(\zeta) - \Psi_q\left(\frac{z_{0_q}}{L}\right)\right]} \tag{27}$$

$$\tag{28}$$

can be derived by integrating the universal functions (Businger et al., 1971; Dyer, 1974) where

$$\Psi_m(\zeta) = \begin{cases} 2\ln\left(\frac{1+\chi}{2}\right) + \ln\left(\frac{1+\chi^2}{2}\right) - 2tan^{-1}\chi + \frac{\pi}{2} & \zeta < 0 \\ -b\zeta & 0 \leq \zeta \leq 1 \\ (1-b)(1+\ln\zeta) - \zeta & \zeta > 1 \end{cases}, \tag{29}$$

$$\Psi_t(\zeta) = \Psi_q(\zeta) = \begin{cases} \ln\left(\dfrac{1+\chi^2}{2}\right) & \zeta < 0 \\ -b\zeta & 0 \le \zeta \le 1 \\ (1-b)(1+\ln\zeta) - \zeta & \zeta > 1 \end{cases}, \tag{30}$$

with $\chi = (1 - a\zeta)^{1/4}$, $a = 16$ and $b = 5$ are the stability-dependent correction functions. The computation of the stability functions requires an a priori assumption (Munro, 1989) about ~~the~~ L which in turn depends on q_{sh} and the friction velocity

$$u_* = \frac{\kappa u_{z_v}}{\ln\left(\dfrac{z}{z_{0_v}}\right) - \Psi_m(\zeta)}. \tag{31}$$

COSIPY uses an iterative approach to resolve the dependency of these variables. At the beginning of the first iteration u_* (Eq. 31) and q_{sh} (Eq. 17) are approximated assuming a neutral stratification ($\zeta = 0$). These quantities are then used to calculate L (Eq. 24). In the next iteration, the updated L is then used to correct u_* and q_{sh} . The iteration is repeated until either the changes in q_{sh} are less than $\cancel{1e-2}$ $1 \cdot 10^{-2}$ or a maximum number of 10 iterations is reached. As already shown by other studies, the algorithm usually converges in less than 10 time steps (Munro, 1989).

3.5 Snow densification

Snow ~~grain~~ settling during metamorphism and compaction under the weight of the overlying snowpack generally increases the snow density over time ~~(Anderson, 1976; Boone, 2004)~~(Anderson, 1976; Boone, 2004; Essery et al., 2013). The snow density is a key characteristic of the snowpack, which is used by COSIPY to derive important snow properties such as thermal conductivity and liquid water content. Assuming that a rapid settlement of fresh snow occurs simultaneously with slow compaction by the load resisted by the viscosity, the rate of density change of each snow layer becomes

$$\frac{1}{\rho_s(z,t)} \frac{d\rho_s(z,t)}{dt} = \frac{M_s(z,t)\,g}{\eta(z,t)} + c_1 \exp\left[-c_2(T_m - T_s) - c_3 \max(0, \rho_s(z,t) - \rho_0)\right], \tag{32}$$

with M_s is the overlying snow mass, $c_1 = 2.8 \times 10^{-6}\,\mathrm{s^{-1}}$, $c_2 = 0.042\,\mathrm{K^{-1}}$, $c_3 = 0.046\,\mathrm{m^3\,kg^{-1}}$, and the viscosity

$$\eta(z,t) = \eta_0 \exp\left[c_4(T_m - T_s) + c_5\rho_s\right] \tag{33}$$

where $\eta_0 = 3.7 \times 10^7\,\mathrm{kg\,m^{-1}\,s^{-1}}$, $c_4 = 0.081\,\mathrm{K^{-1}}$, and $c_5 = 0.018\,\mathrm{m^3\,kg^{-1}}$ ~~(Anderson, 1976; Boone, 2004)~~(Anderson, 1976; Boone, 2
.

3.6 Mass changes

The total mass changes may be written as the integral expression

$$\frac{\partial}{\partial t}\int_0^d \rho_s\,dz = \frac{\partial}{\partial t}\int_0^d \theta_i(z,t)\rho_i\,dz + \frac{\partial}{\partial t}\int_0^d \theta_w(z,t)\rho_w\,dz + \frac{\partial}{\partial t}\int_0^d \theta_a(z,t)\rho_a\,dz, \tag{34}$$

which follows directly from Eq. (2). So any net mass change must be accompanied by changes in ice fraction, liquid water content, and porosity within the snow/ice column of height d. The continuity equation for ice fraction (first term on the right side) may be written as

$$\frac{\partial}{\partial t}\int_0^d \theta_i(z,t)\rho_i\,dz = \frac{\partial}{\partial t}\int_0^d \Delta\theta_i(z,t)\rho_i\,dz + SF - \frac{q_m}{L_f} + \frac{q_{lh}}{L_s} + \frac{q_{lh}}{L_v}, \tag{35}$$

where the integral on the right side describes the internal mass changes by melt and refreezing, SF the mass gain by snowfall, and q_m/L_f is the mass loss by melt. The last two terms of this equation, q_{lh}/L_s and q_{lh}/L_v, are the sublimation/deposition and evaporation/condensation fluxes at the surface, respectively, depending on the sign of q_{lh} and $T_s(z=0,t)$. Melt energy q_m is the energy surplus at the surface which is available for melt, and follows from Eq. (4). Similarly, we can extend the continuity equation for the liquid water content which reads as

$$\frac{\partial}{\partial t}\int_0^d \theta_w\rho_w(z,t)\,dz = \frac{\partial}{\partial t}\int_0^d \Delta\theta_w(z,t)\rho_w\,dz + R_f + \frac{q_m}{L_f} + \frac{q_{lh}}{L_v} - Q \tag{36}$$

with the integral on the right side describing the internal mass changes of liquid water by melt and refreezing, R_f the mass gain by liquid precipitation, and Q the runoff at the bottom of the snowpack. COSIPY calculates all terms and writes them to the output file.

4 Model architecture

Basically, ~~the~~ COSIPY ~~model~~ consists of a model kernel which is extended by modules. The model kernel forms the underlying model structure and provides the IO routines, takes over the discretisation of the computational mesh, parallelizes the simulations, and solves the fundamental mass- and energy conservation equations Eq. (2) and Eq. (3). These tasks are independent of the implementations of the parametrization and usually, do not require any modification by the end-user.

COSIPY is a one-dimensional model that resolves vertical processes at a specific point on the glacier. For spatially distributed simulations, the point model is integrated independently at each point of the glacier domain, neglecting the lateral mass and energy fluxes. The independency of the point models simplifies scaling for larger computer architectures, which led to the COSIPY model architecture being designed for both local workstations and High-Performance Computing Cluster (HPCC). So far, the model has been successfully tested on Slurm Workload Manager (https://slurm.schedmd.com) and PBS job scheduling systems (https://www.pbspro.org). Regardless of whether the distributed simulations are integrated on a single-core or multi-core computing environment, the point model sequence is always the same. During initialisation, the atmospheric input data is read in, and the ~~dynamic~~ mesh is generated. With distributed spatial simulations, the data is distributed across the available cores, and one-dimensional calculations are performed for each grid point.

At the beginning of each time step, it is checked whether snowfall occurs and must be added to the existing snow cover. Subsequently, the computational mesh is re-meshed to ensure numerical stability. Afterwards, the albedo (Eq. 11) and the roughness length are updated. ~~Afterwards, the computational mesh is re-meshed to ensure numerical stability.~~ Once these steps have been performed, the heating and melting of snow by penetrating short-wave radiation (Eq. 13, 6, and 7) is determined and the surface energy fluxes and surface temperature (Eq. 4) are derived. The resulting meltwater, both from surface and subsurface melt, is then percolated through the layers (bucket approach). Next the heat equation (Eq. 3) is solved after all terms on the right side have been determined.

4.1 Input and Output (IO) and initial condition

The model is driven by meteorological data that must be provided in a corresponding NetCDF file (see https://cosipy.readthedocs.io/en/latest/Ressources.html). Input parameters include atmospheric pressure, air temperature, cloud cover fraction, relative humidity, incoming shortwave radiation, total precipitation and wind velocity. Optional snowfall and incoming longwave radiation can be used as forcing parameters. If the snow height (or snow water equivalent) and/or surface temperature are also specified in the input file, these are used as initial conditions. Otherwise, snow depth and surface temperature are assumed to be homogeneous in space at the start of the simulation according to the specifications in the configuration file. In addition to meteorological parameters, COSIPY requires static information such as topographic parameters and a glacier mask. ~~Various tools are available to create and convert the~~ Example workflows for creating and converting static and meteorological data into ~~a corresponding input file. An example workflow for creating~~ the required NetCDF input is included in the source code (https://cosipy.readthedocs.io/en/latest/Documentation.html#quick-tutorial). Besides the standard output variables, there is also the possibility to store vertical snow profile information~~(not recommended for distributed simulations)~~, although to save memory we can only recommend this for single point simulations. To reduce the amount of data, the users can specify which of the output variables will be stored.

5 ~~Example - Zhadang glacier, High mountain Asia~~Model applications

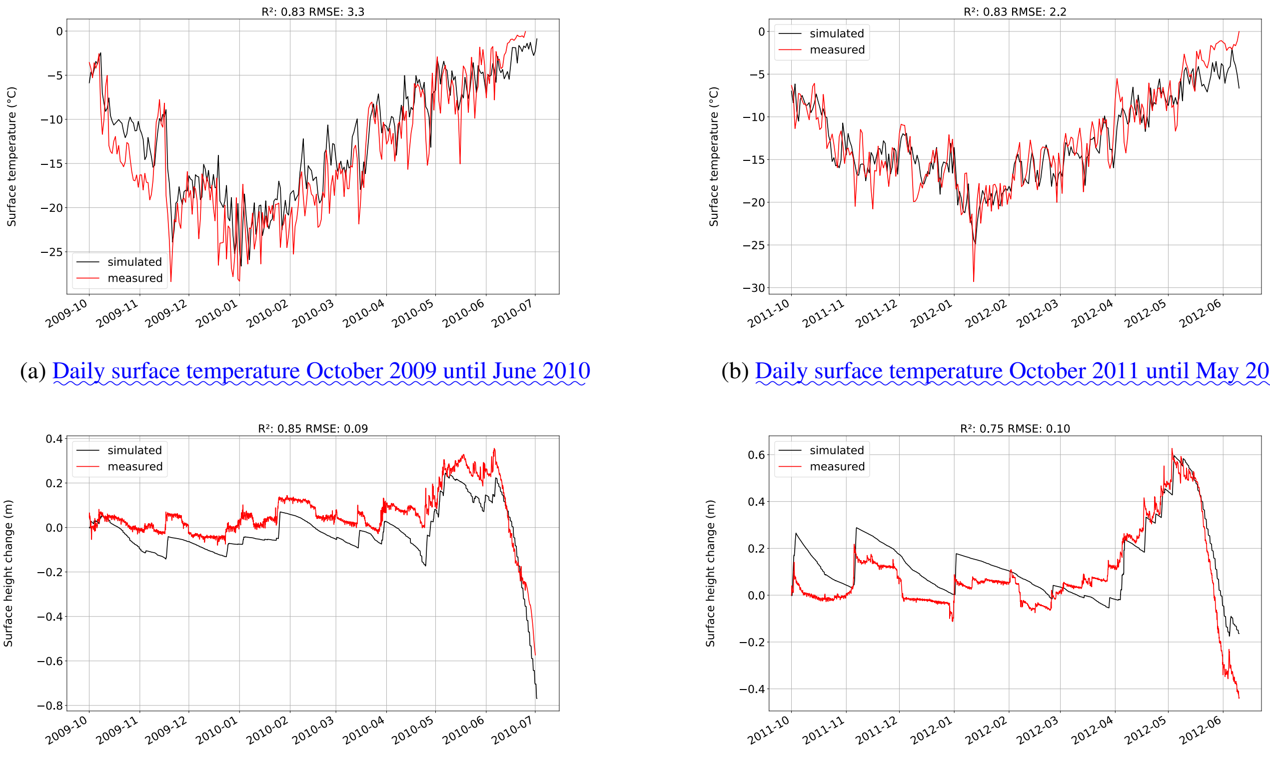
~~To illustrate a model application , we show a mass- and energy balance simulation of the Zhadhang~~

5.1 Zhadang glacier, High Mountain Asia

The first example shows the application of COSIPY to the Zhandang glacier, which is located on the north-eastern slope of the Nyainqentanglha Mountains (30°28.2’N, 90°37.8’E) on the Central Tibetan Plateau. ~~We~~

5.1.1 Single-site simulation

For single-site simulation, we use hourly data from May 2009 to June 2012 from an automatic weather station (AWS) on the Zhadhang Glacier (Huintjes et al., 2015b). The relevant variables air pressure p_{z_t}, air temperature T_{z_t}, relative humidity RH_{z_t}, incident short-wave radiation q_G, snowfall SF and wind speed u_{z_v} were measured by the AWS. Due to the harsh and remote environment, the time series show gaps that were filled with the High Asia Refined Analysis (HAR; Maussion et al., 2014) product. The cloud cover fraction N was provided by ERA5 (Hersbach and Dee, 2016) data. We compare the simulated snow temperature T_s and surface height change ΔH with the AWS measurements. Furthermore, ablation stakes were installed on the glacier to determine the loss of mass at various locations on the glacier. A detailed description of the data, the AWS sensors used, the post-processing procedure and the discussion can be found in Huintjes et al. (2015b) and Huintjes (2014).



(a) Daily surface temperature October 2009 until June 2010

(b) Daily surface temperature October 2011 until May 2012

(c) Hourly surface height change October 2009 until June 2010

(d) Hourly surface height change October 2011 until May 2012

Figure 1. Simulated and measured surface temperatures and surface height changes (in both cases permanent snow cover) at the location of the automatic weather station at the Zhadang glacier.

Table 1. Observed and simulated ice ablation (mm w.e.) for three periods at the automatic weather station on the Zhadang glacier

Period	13.07.2009-30.08.2009		17.05.2010-10.09.2010		26.07.2011-16.08.2011	
	total	per day	total	per day	total	per day
Stake	1072	22	2255	19	150	7
Simulated	1190	25	2150	19	160	8

Simulation. Figure 1a and 1b show the glacier surface temperatures determined from longwave radiation measurements and from COSIPY simulations for two periods where in-situ measurements ~~were~~ are available. The model represents both the daily variability ($R^2 = 0.83$, ~~p-value~~ p-values < 0.001) and the magnitude of the observed surface temperature. The root mean square error is ~~2.9 K and 2.7~~ 3.3 K and 2.2 K for the two periods, respectively, and is thus within the typical uncertainty range of long-wave radiation measurements. The ~~modelled~~ simulated cumulative mass balance over the entire period from ~~October~~ April 2009 to May 2012 is ~~-6.71~~-2.9 m w.e. ~~The most negative ablation stake shows a negative surface mass balance of -6.4 m w.e, and thus the modelleld is slightly more negative over the entire study period.~~ Figure 1c and 1d show the ~~modelled~~ simulated and measured ΔH for the two periods October 2009 to June 2010 and October 2011 to May 2012 where measurements are available. The daily and seasonal variability is well captured by the model (R^2 = ~~0.78~~0.85, p-value < 0.001 and R^2 = ~~0.61~~0.75, p-value < 0.001), even if snowfall seems to be too low during the ~~winter months~~first period. Nevertheless, overall the differences are consistently small with ~~RMSE of 0.07~~a RMSE of 0.09 m and ~~0.11~~0.10 m. ~~The~~ Table 1 shows the observed and simulated ice ablation for three different periods for which measurements are available. For all three periods, a high degree of agreement is evident, which reveals that the energy fluxes are represented by COSIPY.

5.1.2 Distributed simulation, scalability

For a distributed glacier-wide ~~cumulative surface mass balance for the decade 2009 to 2018 is presented in Figure 2. The model was driven~~ run we drive COSIPY by ERA5 data instead of in-situ observations. The ERA5 temperature and humidity data were interpolated across the topography using empirical lapse rates. Atmospheric pressure has been corrected using the barometric formula. The radiation model of Wohlfahrt et al. (2016) was used for the incoming shortwave radiation to account for effects of shadowing, slope and aspect. Total precipitation, cloud cover and horizontal wind velocity were used directly from the closest ERA5 grid point (cf. Table 2). The computational domain consisted of 1837 grid cells with a spatial resolution of approximately 30 m (1 arcsecond) (see Fig. 2). ~~The modelled total mass loss during this period was $-13.9\ m\ w.e.$. The distributed model shows a lower mean annual mass loss ($-0.7\ m\ w.e.\ a^{-1}$, only May~~

Table 2. COSIPY forcing variables and applied downscaling approaches for distributed simulation.

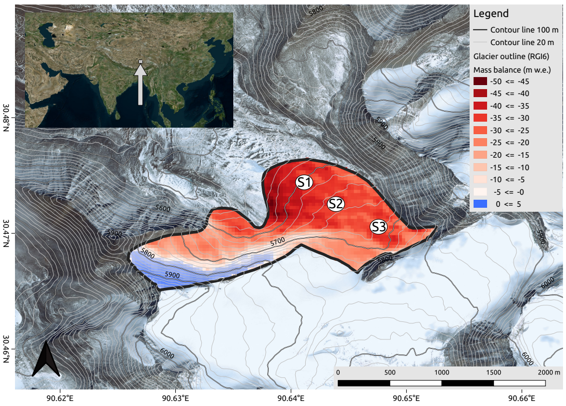
Variable	Downscaling ERA5 data to elevation of the glacier	Applied approach for distributed fields on the glacier
Air pressure p_{z_t}	Barometric formula	Barometric formula
Air temperature T_{z_t}	Lapse rate	Lapse rate
Cloud cover fraction N	-	-
Incoming shortwave radiation q_G	-	Radiation modelling (Wohlfahrt et al., 2016)
Relative humidity RH_{z_t}	Lapse rate	Lapse rate
Total precipitation RRR	-	-
Wind speed u_{z_v}	-	-

Simulation. The glacier-wide cumulative surface mass balance for the decade 2009 ~~to June 2012 considered) than the point model (-2.6 $m\ w.e.\ a^{-1}$) driven by the in-situ observations from May 2009~~ to ~~June 2012 at the position of the AWS~~2018 is presented in Figure 2. The simulated annual mass balance of for this period was -1.9 $m\ w.e.a^{-1}$. The results are in line with the analysis of Qu et al. (2014) who reported negative mass balances of -1.9, -2.0, -0.8 and -2.7 $m\ w.e$ for the years 2009 to 2012.

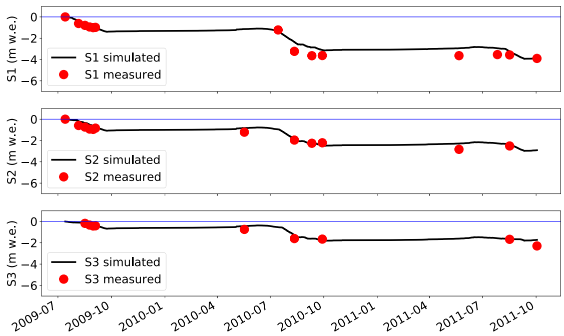
Furthermore, COSIPY reproduced the spatial distribution at different locations in the ablation area of the Zhadang glacier (cf. S1, S2 and S3 in Figure 2b)

Scalability. A big challenge for large applications is usually the computational cost. To achieve the required performance, models should be scalable on parallel high-performance computing environments. For the model performance analysis, we use a cluster with identical nodes, each consisting of two Intel Xeon(R) E5-2640 v4 CPUs operating at 2.4 GHz and connected via InfiniBand. Each processor has ten cores, 32 GB memory and a memory bandwidth of 68.3 GB/s. To test the performance of the parallelized COSIPY version, we performed a spatial simulation of the Zhadhang glacier. We used a 3 arcsecond ($\sim$ 90 m) Shuttle Radar Topography Mission (SRTM) terrain model so that the computational grid consists of 206 points. The performance of the parallel version was then compared to the single-core solution by measuring the required execution time for different core setups (1-220 cores). Figure 3 shows the speedup compared to the single-core version, i.e. the ratio of the original execution time (single core) with the execution time of the corresponding node test (multiple cores). If the model is executed with 20 cores, the speedup is ~ 2. With 120 cores a speedup of ~ 10 is reached, i.e. each core has to calculate a maximum of two grid points. A speedup of more than ~ 16 is not possible with this system and is achieved with a number of 220 cores (more cores than grid points). The computation time is less than 35 minutes for a ten year period (hourly resolution) when using 220 cores. At this point, it should be mentioned that the performance can vary significantly on other HPCC systems and simulation conditions and should always be checked before submitting larger simulations to the cluster.

~~Model frameworks for calculating the climatic surface mass balance of glaciers based on energy exchange at the surface of snow and ice have been available since the last decades of~~



(a) ~~Daily surface temperature October 2009 until June 2010~~



(b) ~~Daily surface temperature October 2011 until May 2012~~

Figure 2. ~~Hourly surface height change October~~ Distributed mass balance simulation of the Zhadang glacier. **(2a)** Cumulative climatic mass balance from 2009 ~~until June 2010~~to 2018 with 1827 grid points, contour lines (SRTM), glacier outline from Randolph Glacier Inventory 6.0 and a topographic map from Bing Maps (Microsoft, 2020); **(2b)** comparison of three measurements (ablation stakes) from July 2009 to October 2011 with the simulated cumulative surface mass balance of the corresponding grid point.

~~Hourly surface height change October 2011 until May 2012 Modelled and measured surface temperatures and height changes at the location of the at the Zhadang glacier.~~

~~Cumulative climatic surface mass balance from 2009 to 2018 Zhadang glacier with 1837 grid points, contour lines (), glacier outline from and a topographic map from Bing Maps (Microsoft, 2020).~~

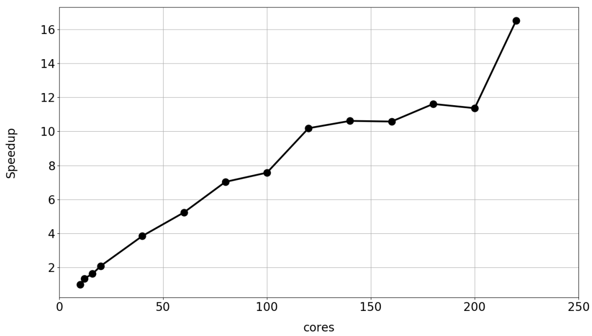


Figure 3. Speedup (execution time of single-core simulation divided by execution time of the corresponding multiple-core simulation) for computing a 10-year distributed COSIPY run on Zhadang glacier with 206 grid points.

5.2 Distributed mass- and energy balance simulation and operational application at Hintereisferner in the Austrian Alps

The Hintereisferner (HEF) is a valley glacier located in the Ötztal Alps of Austria (46.79°N, 10.74°E). The glacier begins high on the ~~20th Century. The approaches commonly use either meteorological observations or modelled atmospheric data. The energy available for melting snow and ice mostly results as the residual term of~~ flank of the ~~energy balance equation. The energy balance~~ mountain Weißkugel, at approximately 3720 m, and runs down to its terminus at approximately 2460 m. HEF is a prime location for meteorological and glaciological research activities due to its monitoring infrastructure. There is a network of 4 automatic weather stations (AWS) and 4 precipitation gauges operated on, and in the vicinity of HEF. Since 2016, the University of Innsbruck is also running a permanent Terrestrial Laser Scanner (TLS) and a 5 m meteorological flux tower. Measurement data is hourly transmitted to a data server. COSIPY is now being used to develop an operational mass balance prediction system for the 'Hintereisferner'. The model is driven by the latest COSMO2 analysis and forecast data. With the forecast data the energy and mass flows on the glacier are predicted for the next 24 hours with a horizontal resolution of 30 m. The simulated fields are automatically visualised and provided on a web server. In the future the TLS measurements will be used to improve the forecast continuously. The system is currently running in test mode but will be available to the public in the next months.

In addition to the energy and mass flows at the surface~~and~~, the snow/ice profiles will be stored. This will allow to compare the results with snow pits and to test the implementation of different parameterizations.

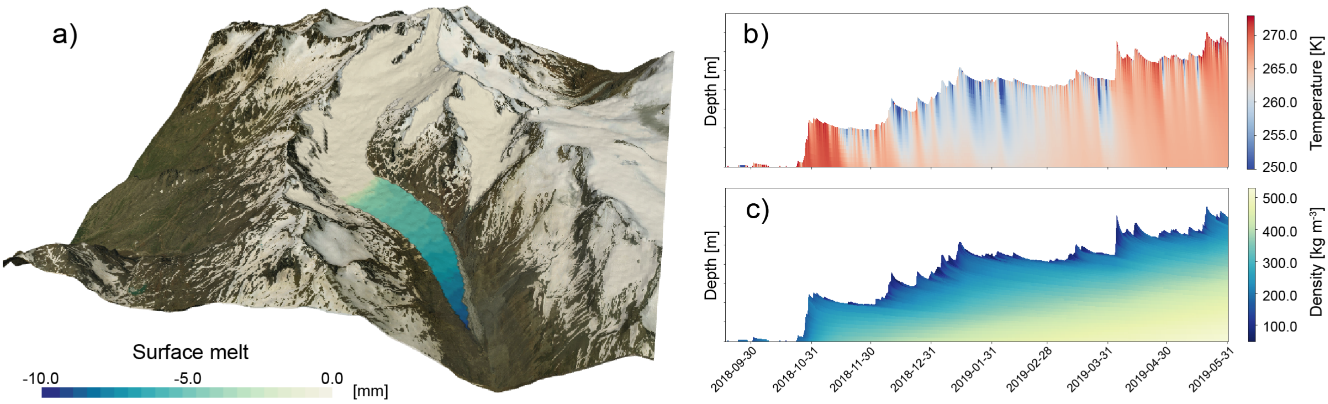
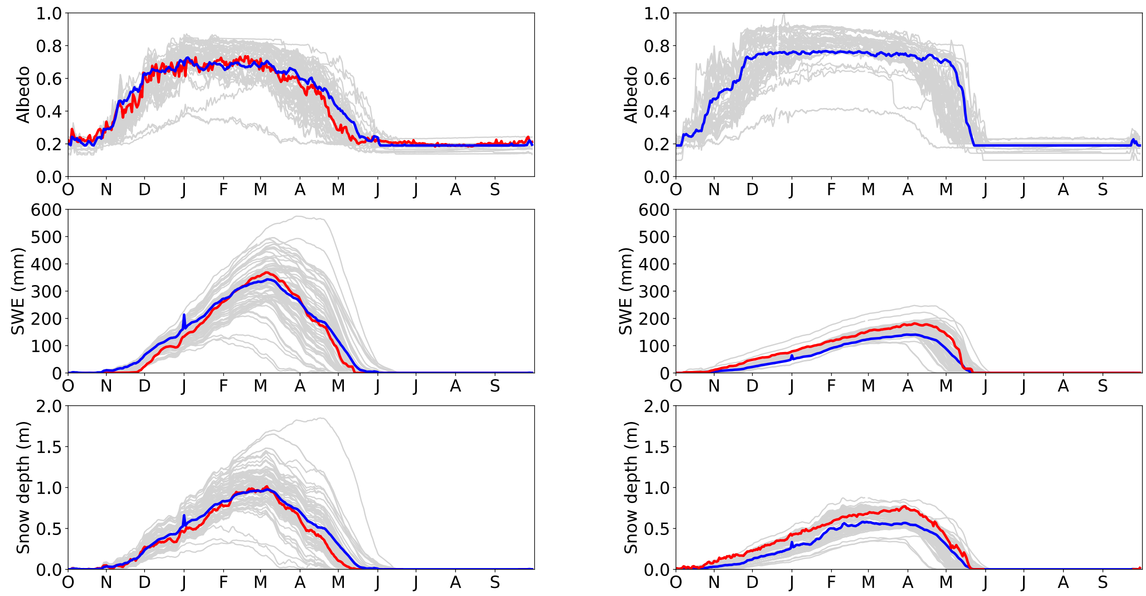


Figure 4. Operational application of COSIPY for the Hintereisferner. Panel a) shows the forecast of surface melt for 22 June 2020 based on COSMO2 data. Panels b) and c) show an example of the temperature and density profile for one site on the glacier tongue for the period September 2018 to June 2019.

5.3 Model intercomparison - Earth System Model-Snow Model Intercomparison Project

Within the Earth System Model-Snow Model Intercomparison Project (ESM-SnowMIP, Krinner et al., 2018) several of snow models were compared to evaluate different snow schemes and to improve the coupling of land surface snow models in Earth System models. Ménard et al. (2019) describes the standardized input and evaluation data. Ten different sites representing



(a) Col de porte, France (1994-2014) (b) Sodankylä, Finland (1997-2014)

Figure 5. Comparison of long-term daily mean COSIPY with ESM-SnowMIP simulations for two sites. COSIPY simulations (blue lines), measurements (red lines) and ESM-SnowMIP (grey lines) simulations of albedo, snow water equivalent (SWE) and snow depth at two sites. Measurements and simulations provided by Krinner et al. (2018).

mountainous regions (Europe and western USA), boreal forests (Canada), the Arctic (Finland) and urban regions (Japan) for periods between seven and 20 years (hourly resolution) are provided, including meteorological classification and details on measuring instruments and data processing. These quality controlled data are freely available on a PANGAEA repository (Ménard and Essery, 2019) and provide the possibility to benchmark new model developments, to detect uncertainties and to reduce model errors. Unlike most of the models participating in the ~~energy fluxes within the snow or ice volume are coupled via the surface properties, in particular the surface skin temperature. Since snow and ice surfaces cannot exceed the melting point temperature, any energy excess must be transferred to the phase change from solid to liquid water, respectively snow or ice melt.~~ Intercomparison project, COSIPY ~~is an open-source and fully documented community model of its kind~~is not a pure snow model, but still all necessary forcing variables are available to apply the model to the different test data sets. We downscaled wind speed from 10 to 2 m above ground using the logarithmic wind profile and calculated the relative humidity from the specific humidity using the saturation mixing ratio and water vapour. The simulated abledo, snow water equivalent (SWE) and snow depth were compared with the evaluation data offered on the online repository (https://doi.pangaea.de/10.1594/PANGAEA.897575). Surface and soil temperature could not be compared, because no soil scheme is implemented in COSIPY which allows for warm surface and underground temperatures above the melting point. Figure 5 shows the daily long-term mean values of albedo, SWE and snow depth for two example sites. The abledo parametrization was calibrated to fit the observed values at Col the porte. With the calibrated albedo parameterisation, COSIPY can reproduce the observed long-term snowpack evolution. The results for Sodankylä, Finnland (cf. 5b) show a little lower snowpack compared to the measurements. The COSIPY runs for both sites are in the range of the ESM-SnowMIP ensemble simulations (see Figure 5, Krinner et al., 2018).

6 Conclusions

COSIPY provides a lean, flexible and user-friendly framework for modelling distributed snow and glacier mass changes. It provides a suitable platform for sensitivity, detection and attribution analyses as well as a tool for the quantification of inherent uncertainties in mass balance studies. The model has a modular structure ~~facilitating~~ and allows the exchange of routines or ~~parametrizations for individual processes with only~~ parameterizations of individual physical processes with little effort. This structure allows the end user to quickly adapt the model to their needs. The open design of COSIPY ~~, its well-documented and its~~ is well documented, and the modular approach allows ~~for~~ a joint community-driven ~~advancement~~ further development of the model in the future. In order to increase user-friendliness, additional functions are available, such as a restart option for operative applications and an automatic comparison between simulation and ablation data. These functions will be further refined in the future.

~~may serve to analyse inter-annual and intra-annual variations of energy and surface mass balance of glaciers. Further, it allows identifying sensitivities, non-linearities, co-variances, and tipping points in the components of glacier surface energy and mass fluxes resulting from the variability of atmospheric forcing. Since it is notoriously difficult to obtain sufficiently reliable atmospheric forcing data, the uncertainties in climatic mass balance estimates derived only from modelling can be quite high. Therefore, it is recommended to use observations of mass balance either from fieldwork or from remote sensing data analysis to benchmark model results whenever possible~~The model is written in Python and completely based on open source libraries. The model, source code, and examples are provided on a freely accessible Git repository (https://github.com/cryotools/cosipy) for non-profit purposes. The aim is to set up a community platform where scientists can actively participate in extending and improving the model code. To ensure quality control of the model code, changes to the code are automatically tested with Travis CI (www.travis-ci.org) when they are uploaded to the repository. It is planned to release updates at regular intervals. To make working with COSIPY easier, a community platform (https://cosipy.slack.com) has been set up in addition to readthedocs documentation (https://cosipy.readthedocs.io/en/latest), which allows users and developers to share experiences, report bugs and communicate needs.

Future improvements of COSIPY are expected by applying the model in different climates and varying topographical settings. Additional processes affecting the climatic mass balance of glaciers such as debris cover ~~, snowdrift and avalanches~~ and snowdrift can be considered in further developments of the model. On the long run, one of the priorities will be to create a multiphysics environment that allows ensemble runs. In principle it is already possible to create ensemble simulations with different physical parameterizations and solvers, but COSIPY is not yet an ensemble multiphysics modelling environment. As a vision for the future it is conceivable to extend COSIPY for automatic ensemble simulations. So far, it is only possible to run COSIPY with different combinations of physical parameterizations or input uncertainties and then evaluate the statistics.

7 Code availability, documentation, and software requirements

COSIPY is based on the Python 3 language and is provided on a freely accessible git repository (https://github.com/cryotools/cosipy, last access: ~~January~~ June 20, 2020). COSIPY can be used for non-profit purposes under the GPLv3 license (http://www.gnu.org/licenses/gpl-3.0.html). Scientists can actively participate in model development. A documentation with a sample workflow, information about input/output formats and the code structure is available under 'Read the Docs' (https://cosipy.readthedocs.io/en/latest/index.html, last access: ~~January~~ June 20, 2020). As a community platform and user support, we use the groupware Slack (https://cosipy.slack.com, last accessed: ~~January~~ June 20, 2020). The various official model releases will be registered with a unique DOI on Zenodo (https://doi.org/10.5281/zenodo.2579668, last access: ~~January~~ June 20, 2020). For the result of this publication the version v1~~.2 (~~.3 (https://doi.org/10.5281/zenodo.3902191) was used. Each commit will be automatically tested with different Python 3 releases on Travis (https://travis-ci.org/cryotools/cosipy, last accessed ~~January~~ June 20, 2020). The tested code coverage is tracked on CodeCov (https://codecov.io/github/cryotools/cosipy/, last accessed ~~January~~ June 20, 2020). Since we have just started writing the tests, code coverage of 35 % is still low but will be increased in the near future. With the exception of the pre-processor for creating the static file (currently not working on Windows systems) the model should work on any operating system with Python 3 installed. However, support for operating systems other than Linux-based systems is limited because we develop and run COSIPY exclusively on Linux-based systems. COSIPY is built on the following open-source libraries: numpy (van der Walt et al., 2011), scipy (Virtanen et al., 2019), xarray ~~(?)~~ (Hoyer and Hamman, 2017), distribued, dask_jobqueue ~~(?)~~(Dask Development Team, 2016), and netcdf4 (https://doi.org/10.5281/zenodo.2669496).

Appendix A: List of symbols

Constant	**Description**	**Unit**	**Default value**
c_p	specific heat of air	$J\ kg^{-1}\ K^{-1}$	1004.67
c_i	specific heat of ice	$J\ kg^{-1}\ K^{-1}$	2050.0
c_w	specific heat of water	$J\ kg^{-1}\ K^{-1}$	4217.0
d^*	albedo depth scale	cm	3.0
g	gravitational acceleration	$m\ s^{-2}$	9.81
k_a	thermal conductivity of air	$W\ m^{-1}\ K^{-1}$	0.026
k_i	thermal conductivity of ice	$W\ m^{-1}\ K^{-1}$	2.25
k_w	thermal conductivity of water	$W\ m^{-1}\ K^{-1}$	0.6089
Pr	turbulent Prandtl number	–	0.8
T_m	melting point temperature	K	273.16
α_s	fresh snow albedo	–	0.9
α_f	firn albedo	–	0.55
α_i	ice albedo	–	0.3
ε_s	surface emissivity	–	0.99
η_0	snow viscosity	$kg\ m^{-1}\ s^{-1}$	3.7×10^7
κ	von Kármán constant	–	0.41
ρ_a	air density	$kg\ m^{-3}$	1.1
ρ_w	water density	$kg\ m^{-3}$	1000.0
ρ_i	ice density	$kg\ m^{-3}$	917.0
ρ_0	snow compaction parameter	$kg\ m^{-3}$	150.0
σ	Stefan–Boltzmann constant	$W\ m^{-2}\ K^{-4}$	5.67×10^8
τ^*	albedo time scale	$days$	22

Variable	**Description**	**Unit**
c_s	specific heat of snow	$J\ kg^{-1}\ K^{-1}$
C_D	bulk transfer coefficient for momentum	–
C_E	bulk transfer coefficient for latent heat	–
C_H	bulk transfer coefficient for sensible heat	–
e_{z_t}	water vapour pressure at height z_t	Pa
Ew_{z_t}	saturation water vapour at height z_t	Pa
$Ew_{z_{0_t}}$	saturation water vapour at the surface	Pa

k_s	thermal conductivity of snow	$W\ m^{-1}\ K^{-1}$
L	Obukhov length	m
L_s	latent heat of sublimation	$J\ kg^{-1}$
L_f	latent heat of fusion	$J\ kg^{-1}$
L_v	latent heat of vaporisation	$J\ kg^{-1}$
ME	available melt energy	$W\ m^{-2}$
M_s	overlying mass	kg
N	cloud cover fraction	$-$
q_{lw}	net longwave radiation	$W\ m^{-2}$
$q_{lw_{in}}$	incoming longwave radiation	$W\ m^{-2}$
$q_{lw_{out}}$	outgoing longwave radiation	$W\ m^{-2}$
q_{sw}	net shortwave radiation	$W\ m^{-2}$
q_{sh}	sensible heat flux	$W\ m^{-2}$
q_{lh}	latent heat flux	$W\ m^{-2}$
q_{rr}	heat flux from rain	$W\ m^{-2}$
q_m	melt energy	$W\ m^{-2}$
q_0	mixing ratio at the surface	$kg\ kg^{-1}$
q_{z_q}	mixing ratio at height z_q	$kg\ kg^{-1}$
p_{z_t}	air pressure at height z_t	hPa
Q	runoff	$mw.e.$
Q_p	volumetric energy sink/source by melting and refreezing	$W\ m^{-3}$
Q_r	volumetric energy surplus by absorption of shortwave radiation	$W\ m^{-3}$
RH_{z_t}	relative humidity at height z_v	$\%$
Ri_b	Bulk Richardson number	$-$
SF	snowfall	m
T_s	snow temperature	K
T_v	virtual air temperature	K
T_{z_t}	air temperature at height z_t	K
T_0	surface temperature	K
u_{z_v}	wind speed at height z_v	$m\ s^{-1}$
u_*	Friction velocity	$m\ s^{-1}$
z_t	measurement height of temperature	m
z_q	measurement height of humidity	m
z_t	measurement height of wind velocity	m
z_{0_v}	aerodynamic roughness length	m

z_{0_t}	roughness length for temperature	m
z_{0_q}	roughness length for specific humidity	m
α	snow/ice albedo	$-$
ε_{cl}	emissivity of clouds	$-$
ε_{cs}	clear sky emissivity	$-$
ε_a	total atmospheric emissivity	$-$
η	snow viscosity	$kg\ m^{-1}\ s^{-1}$
θ_w	liquid water content	$-$
θ_a	air porosity	$-$
θ_i	volumetric ice fraction	$-$
θ_e	irreducible water content	$-$
Θ	local slope	$^\circ$
λ_r	fraction of absorbed radiation	$-$
ρ_s	snow density	$kg\ m^{-3}$
Ψ_{Ri}	stability function based on the Richardson-Number	$-$
Ψ_m	stability function for momentum based on the Monin-Obukhov similarity theory	$-$
Ψ_t	stability function for heat based on the Obukhov length	$-$
Ψ_q	stability function for moisture based on the Obukhov length	$-$

Author contributions. TS and CS are the initiators of the model. TS developed the model design and wrote most of the sections on the physical and numerical principles of the model. AA developed parts of the parameterizations and core classes, applied the model to the Zhadang glacier and the two sites of the ESM-SnowMIP. He also wrote the sections about the corresponding applications and code availability. TS did the simulations for the Hintereisferner. AA and TS were equally involved in the development of the documentation and maintenance of the Community Platform (Slack). During the entire development process, all authors discussed the individual steps of model development, the results and the structure of the manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We gratefully acknowledge financial support by the Deutsche Forschungsgemeinschaft (DFG) with its project 'The impact of the dynamic and thermodynamic flow conditions on the spatio-temporal distribution of precipitation in southern Patagonia' (grant no. SA 2339/4-1) and 'Snow Cover Dynamics and Mass Balance on Mountain Glaciers' (grant no. SA 2339/7-1). Part of this work and the position of co-author Anselm Arndt was financed through the German Research Foundation's (DFG) research grant 'Precipitation patterns, snow and glacier response in High Asia and their variability on sub-decadal time scales, sub-project: snow cover and glacier energy and mass balance variability' (prime-SG, SCHN 680/13-1). The development of the earlier version of the software and the field data used in this study were financed by the projects 'Dynamic response of glaciers of the Tibetan Plateau' (Dyn RG TiP, grant nos. SCHN 680/3-1, SCHN 680/3-2, SCHN 680/3-3) of DFG's Priority Programme 1372 'Tibetan Plateau: Formation—Climate—Ecosystems' (TiP) and the German Federal Ministry of Education and Research's (BMBF) programme 'Central Asia Monsoon Dynamics and Geo-Ecosystems' (CAME), project 'Variability and Trends of Water-Budget Components in Benchmark Catchments of the Tibetan Plateau' (WET, grant no. 03G0804E). We would like to thank Eva Huintjes for her dedication to the fieldwork and for her contributions to the earlier versions of the software. Further, we acknowledge the efforts of all researchers and technicians from the involved German institutions and the Institute of Tibetan Plateau Research of the Chinese Academy of Sciences (CAS) for fieldwork at Zhadang Glacier. We specifically acknowledge the contribution of Yang Wei, Yao Tandong and Kang Shichang in this respect. We gratefully thank Richard Essery and Gerhard Krinner for providing the data of the ensemble simulations for two sites of ESM-SnowMIP. We also wish to thank David Loibl for his contribution in the form of logo, ideas and discussions regarding programming strategies, and for hosting COSIPY on his platform https://cryo-tools.org/. We would also like to thank Samuel Morin and the anonymous reviewer for their constructive comments and ideas.

References

Anderson, E. A.: Development and testing of snow pack energy balance equations, Water Resources Research, 4, 19–37, https://doi.org/10.1029/WR004i001p00019, http://doi.wiley.com/10.1029/WR004i001p00019, 1968.

Anderson, E. A.: A point energy and mass balance model of a snow cover, Technical Report, National Weather Service (NWS), United States, 1976.

Bartelt, P. and Lehning, M.: A physical SNOWPACK model for the Swiss avalanche warning: Part I: numerical model, Cold Regions Science and Technology, 35, 123–145, https://doi.org/10.1016/S0165-232X(02)00074-5, https://linkinghub.elsevier.com/retrieve/pii/S0165232X02000745, 2002.

Bintanja, R. and Van Den Broeke, M. R.: The Surface Energy Balance of Antarctic Snow and Blue Ice, Journal of Applied Meteorology, 34, 902–926, https://doi.org/10.1175/1520-0450(1995)034<0902:TSEBOA>2.0.CO;2, http://journals.ametsoc.org/doi/abs/10.1175/1520-0450%281995%29034%3C0902%3ATSEBOA%3E2.0.CO%3B2, 1995.

Boone, A.: Description du Schema de Neige ISBA-ES (Explicit Snow), Tech. rep., Centre National de Recherches Météorologiques, Météo-France, Toulouse, updated in November, 2009, 2004.

Businger, J. A., Wyngaard, J. C., Izumi, Y., and Bradley, E. F.: Flux-profile relationships in the atmospheric surface layer, Journal of the Atmospheric Sciences, 28, 181–189, https://doi.org/10.1175/1520-0469(1971)028<0181:FPRITA>2.0.CO;2, https://doi.org/10.1175/1520-0469(1971)028<0181:FPRITA>2.0.CO;2, 1971.

Cogley, J. G., Hock, R., A Rasmussen, L., Arendt, A., Bauder, A., J Braithwaite, R., Jansson, P., Kaser, G., Möller, M., Nicholson, L., and Zemp, M.: Glossary of glacier mass balance and related terms, International Association of Cryospheric Sciences, https://doi.org/10.5167/uzh-53475, 2011.

Coléou, C. and Lesaffre, B.: Irreducible water saturation in snow: experimental results in a cold laboratory, Annals of Glaciology, 26, 64–68, https://doi.org/10.3189/1998AoG26-1-64-68, https://www.cambridge.org/core/product/identifier/S0260305500014579/type/journal_article, 1998.

Conway, J. and Cullen, N.: Constraining turbulent heat flux parameterization over a temperate maritime glacier in New Zealand, Annals of Glaciology, 54, 41–51, https://doi.org/10.3189/2013AoG63A604, https://www.cambridge.org/core/product/identifier/S0260305500260588/type/journal_article, 2013.

Dask Development Team: Dask: Library for dynamic task scheduling, DASK, https://dask.org, 2016.

Dyer, A. J.: A review of flux-profile relationships, Boundary-Layer Meteorology, 7, 363–372, https://doi.org/10.1007/BF00240838, http://link.springer.com/10.1007/BF00240838, 1974.

Essery, R., Morin, S., Lejeune, Y., and B Ménard, C.: 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, https://linkinghub.elsevier.com/retrieve/pii/S0309170812002011, 2013.

Ferziger, J. H. and Perić, M.: Computational Methods for Fluid Dynamics, Springer Berlin Heidelberg, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-56026-2, http://link.springer.com/10.1007/978-3-642-56026-2, 2002.

Fletcher, R.: Practical Methods of Optimization, John Wiley & Sons, Ltd, Chichester, West Sussex England, https://doi.org/10.1002/9781118723203, http://doi.wiley.com/10.1002/9781118723203, 2000.

Foken, T.: Micrometeorology, Springer Berlin Heidelberg, Berlin, Heidelberg, https://doi.org/10.1007/978-3-540-74666-9, http://link.springer.com/10.1007/978-3-540-74666-9, 2008.

Galos, S. P., Klug, C., Maussion, F., Covi, F., Nicholson, L., Rieg, L., Gurgiser, W., Mölg, T., and Kaser, G.: Reanalysis of a 10-year record (2004-2013) of seasonal mass balances at Langenferner/Vedretta Lunga, Ortler Alps, Italy, The Cryosphere, 11, 1417–1439, https://doi.org/10.5194/tc-11-1417-2017, http://www.the-cryosphere.net/11/1417/2017/, 2017.

Gurgiser, W., Marzeion, B., Nicholson, L., Ortner, M., and Kaser, G.: Modeling energy and mass balance of Shallap Glacier, Peru, The Cryosphere, 7, 1787–1802, https://doi.org/10.5194/tc-7-1787-2013, https://www.the-cryosphere.net/7/1787/2013/, 2013.

Hantel, M., Ehrendorfer, M., and Haslinger, A.: Climate sensitivity of snow cover duration in Austria, Int. J. Climatol., p. 26.

Hersbach, H. and Dee, D.: ~~ERA~~ERA5 ~~5~~ reanalysis is in production, ECMWF Newsletter, Tech. rep., European Centre for Medium-Range Weather Forecasts, https://www.ecmwf.int/en/newsletter/147/news/era5-reanalysis-production, vol. 147, 2016.

Hock, R. and Holmgren, B.: A distributed surface energy-balance model for complex topography and its application to Storglaciären, Sweden, Journal of Glaciology, 51, 25–36, https://doi.org/10.3189/172756505781829566, https://www.cambridge.org/core/product/identifier/S0022143000215013/type/journal_article, 2005.

Hoyer, S. and Hamman, J. J.: xarray: N-D labeled Arrays and Datasets in Python, Journal of Open Research Software, 5, 10, https://doi.org/10.5334/jors.148, http://openresearchsoftware.metajnl.com/articles/10.5334/jors.148/, version: https://zenodo.org/record/3547868#.XhRHl66YUUE, 2017.

Huintjes, E.: Energy and mass balance modelling for glaciers on the Tibetan Plateau - Extension, validation and application of a coupled snow and energy balance model, Ph.D. thesis, Rheinisch-Westfälischen Technischen Hochschule Aachen, Aachen, https://publications.rwth-aachen.de/record/459462/files/5239.pdf, 2014.

Huintjes, E., Neckel, N., Hochschild, V., and Schneider, C.: Surface energy and mass balance at Purogangri ice cap, central Tibetan Plateau, 2001–2011, Journal of Glaciology, 61, 1048–1060, https://doi.org/10.3189/2015JoG15J056, https://www.cambridge.org/core/product/identifier/S0022143000200245/type/journal_article, 2015a.

Huintjes, E., Sauter, T., Schröter, B., Maussion, F., Yang, W., Kropáček, J., Buchroithner, M., Scherer, D., Kang, S., and Schneider, C.: Evaluation of a Coupled Snow and Energy Balance Model for Zhadang Glacier, Tibetan Plateau, Using Glaciological Measurements and Time-Lapse Photography, Arctic, Antarctic, and Alpine Research, 47, 573–590, https://doi.org/10.1657/AAAR0014-073, http://www.bioone.org/doi/10.1657/AAAR0014-073, ~~2015.~~ 2015b.

Klok, E. and Oerlemans, J.: Model study of the spatial distribution of the energy and mass balance of Morteratschgletscher, Switzerland, Journal of Glaciology, 48, 505–518, https://doi.org/10.3189/172756502781831133, https://www.cambridge.org/core/product/identifier/S0022143000209623/type/journal_article, 2002.

Kraus, H.: An energy balance model for ablation in mountainous areas, IAHS Publication, p. 9, 1975.

Krinner, G., Derksen, C., Essery, R., Flanner, M., Hagemann, S., Clark, M., Hall, A., Rott, H., Brutel-Vuilmet, C., Kim, H., Ménard, C. B., Mudryk, L., Thackeray, C., Wang, L., Arduini, G., Balsamo, G., Bartlett, P., Boike, J., Boone, A., Chéruy, F., Colin, J., Cuntz, M., Dai, Y., Decharme, B., Derry, J., Ducharne, A., Dutra, E., Fang, X., Fierz, C., Ghattas, J., Gusev, Y., Haverd, V., Kontu, A., Lafaysse, M., Law, R., Lawrence, D., Li, W., Marke, T., Marks, D., Ménégoz, M., Nasonova, O., Nitta, T., Niwano, M., Pomeroy, J., Raleigh, M. S., Schaedler, G., Semenov, V., Smirnova, T. G., Stacke, T., Strasser, U., Svenson, S., Turkov, D., Wang, T., Wever, N., Yuan, H., Zhou, W., and Zhu, D.: ESM-SnowMIP: assessing snow models and quantifying snow-related climate feedbacks, Geoscientific Model Development, 11, 5027–5049, https://doi.org/10.5194/gmd-11-5027-2018, https://www.geosci-model-dev.net/11/5027/2018/, 2018.

Kuhn, M.: On the Computation of Heat Transfer Coefficients from Energy-Balance Gradients on a Glacier, Journal of Glaciology, 22, 263–272, https://doi.org/10.3189/S0022143000014258, https://www.cambridge.org/core/product/identifier/S0022143000014258/type/journal_article, 1979.

Kuhn, M.: Micro-Meteorological Conditions for Snow Melt, Journal of Glaciology, 33, 24–26, https://doi.org/10.3189/S002214300000530X, https://www.cambridge.org/core/product/identifier/S002214300000530X/type/journal_article, 1987.

Machguth, H., Paul, F., Hoelzle, M., and Haeberli, W.: Distributed glacier mass-balance modelling as an important component of modern multi-level glacier monitoring, Annals of Glaciology, 43, 335–343, https://doi.org/10.3189/172756406781812285, https://www.cambridge.org/core/product/identifier/S0260305500262174/type/journal_article, 2006.

Machguth, H., Paul, F., Kotlarski, S., and Hoelzle, M.: Calculating distributed glacier mass balance for the Swiss Alps from regional climate model output: A methodical description and interpretation of the results, Journal of Geophysical Research, 114, D19 106, https://doi.org/10.1029/2009JD011775, http://doi.wiley.com/10.1029/2009JD011775, 2009.

Male, D. H. and Granger, R. J.: Snow surface energy exchange, Water Resources Research, 17, 609–627, https://doi.org/10.1029/WR017i003p00609, http://doi.wiley.com/10.1029/WR017i003p00609, 1981.

Maussion, F., Scherer, D., Mölg, T., Collier, E., Curio, J., and Finkelnburg, R.: Precipitation Seasonality and Variability over the Tibetan Plateau as Resolved by the High Asia Reanalysis*~~Supplementary~~, Journal of Climate, 27, 1910–1927, https://doi.org/10.1175/JCLI-D-13-00282.1, http://journals.ametsoc.org/doi/abs/10.1175/JCLI-D-13-00282.1, 2014.

Maussion, F., Gurgiser, W., Großhauser, M., Kaser, G., and Marzeion, B.: ENSO influence on surface energy and mass balance at Shallap Glacier, Cordillera Blanca, Peru, The Cryosphere~~Discussions~~, 9, ~~2999–3053,~~ 1663–1683, https://doi.org/10.5194/tc-9-1663-2015, https://www.the-cryosphere.net/9/1663/2015/, 2015.

Michlmayr, G., Lehning, M., Koboltschnig, G., Holzmann, H., Zappa, M., Mott, R., and Schöner, W.: Application of the Alpine 3D model for glacier mass balance and glacier runoff studies at Goldbergkees, Austria, Hydrological Processes, 22, 3941–3949, https://doi.org/10.1002/hyp.7102, http://doi.wiley.com/10.1002/hyp.7102, 2008.

Microsoft: Bing Maps, https://www.bing.com/maps/, 2020.

Morris, E.: Turbulent transfer over snow and ice, Journal of Hydrology, 105, 205–223, https://doi.org/10.1016/0022-1694(89)90105-4, https://linkinghub.elsevier.com/retrieve/pii/0022169489901054, 1989.

Morris, E. M.: Physics-Based Models of Snow, in: Recent Advances in the Modeling of Hydrologic Systems, edited by Bowles, D. S. and O'Connell, P. E., pp. 85–112, Springer Netherlands, Dordrecht, https://doi.org/10.1007/978-94-011-3480-4_5, http://link.springer.com/10.1007/978-94-011-3480-4_5, 1991.

Munro, D. S.: Surface Roughness and Bulk Heat Transfer on a Glacier: Comparison with Eddy Correlation, Journal of Glaciology, 35, 343–348, https://doi.org/10.3189/S0022143000009266, https://www.cambridge.org/core/product/identifier/S0022143000009266/type/journal_article, 1989.

Munro, D. S.: A surface energy exchange model of glacier melt and net mass balance, International Journal of Climatology, 11, 689–700, https://doi.org/10.1002/joc.3370110610, http://doi.wiley.com/10.1002/joc.3370110610, 1991.

Ménard, C. B. and Essery, R.: ESM-SnowMIP meteorological and evaluation datasets at ten reference sites (in situ and bias corrected reanalysis data), https://doi.org/10.1594/PANGAEA.897575, https://doi.org/10.1594/PANGAEA.897575, publisher: PANGAEA Type: data set, 2019.

Ménard, C. B., Essery, R., Barr, A., Bartlett, P., Derry, J., Dumont, M., Fierz, C., Kim, H., Kontu, A., Lejeune, Y., Marks, D., Niwano, M., Raleigh, M., Wang, L., and Wever, N.: Meteorological and evaluation datasets for snow modelling at 10 reference sites: description of in situ and bias-corrected reanalysis data, Earth System Science Data, 11, 865–880, https://doi.org/10.5194/essd-11-865-2019, https://essd.copernicus.org/articles/11/865/2019/, 2019.

Mölg, T. and Hardy, D. R.: Ablation and associated energy balance of a horizontal glacier surface on Kilimanjaro, Journal of Geophysical Research, 109, D16 104, https://doi.org/10.1029/2003JD004338, http://doi.wiley.com/10.1029/2003JD004338, 2004.

Mölg, T., Cullen, N. J., Hardy, D. R., Kaser, G., and Klok, L.: Mass balance of a slope glacier on Kilimanjaro and its sensitivity to climate, International Journal of Climatology, 28, 881–892, https://doi.org/10.1002/joc.1589, http://doi.wiley.com/10.1002/joc.1589, 2008.

Mölg, T., Cullen, N. J., Hardy, D. R., Winkler, M., and Kaser, G.: Quantifying Climate Change in the Tropical Midtroposphere over East Africa from Glacier Shrinkage on Kilimanjaro, Journal of Climate, 22, 4162–4181, https://doi.org/10.1175/2009JCLI2954.1, http://journals.ametsoc.org/doi/abs/10.1175/2009JCLI2954.1, 2009.

Mölg, T., Maussion, F., Yang, W., and Scherer, D.: The footprint of Asian monsoon dynamics in the mass and energy balance of a Tibetan glacier, The Cryosphere, 6, 1445–1461, https://doi.org/10.5194/tc-6-1445-2012, http://www.the-cryosphere.net/6/1445/2012/, 2012.

Mölg, T., Maussion, F., and Scherer, D.: Mid-latitude westerlies as a driver of glacier variability in monsoonal High Asia, Nature Climate Change, 4, 68–73, https://doi.org/10.1038/nclimate2055, http://www.nature.com/articles/nclimate2055, 2014.

Nicholson, L. I., Prinz, R., Mölg, T., and Kaser, G.: Micrometeorological conditions and surface mass and energy fluxes on Lewis Glacier, Mt Kenya, in relation to other tropical glaciers, The Cryosphere, 7, 1205–1225, https://doi.org/10.5194/tc-7-1205-2013, https://www.the-cryosphere.net/7/1205/2013/, 2013.

Obleitner, F. and Lehning, M.: Measurement and simulation of snow and superimposed ice at the Kongsvegen glacier, Svalbard (Spitzbergen)~~: SUPERIMPOSEDICEONKONGSVEGENGLACIER~~, Journal of Geophysical Research: Atmospheres, 109, n/a–n/a, https://doi.org/10.1029/2003JD003945, http://doi.wiley.com/10.1029/2003JD003945, 2004.

Oerlemans, J.: Glaciers and climate change, A.A. Balkema Publishers, Lisse; Exton, (PA), 2001.

Oerlemans, J. and Knap, W. H.: A 1 year record of global radiation and albedo in the ablation zone of Morteratschgletscher, Switzerland, Journal of Glaciology, 44, 231–238, https://doi.org/10.1017/S0022143000002574, https://www.cambridge.org/core/product/identifier/S0022143000002574/type/journal_article, 1998.

Qu, B., Ming, J., Kang, S.-C., Zhang, G.-S., Li, Y.-W., Li, C.-D., Zhao, S.-Y., Ji, Z.-M., and Cao, J.-J.: The decreasing albedo of the Zhadang glacier on western Nyainqentanglha and the role of light-absorbing impurities, Atmospheric Chemistry and Physics, 14, 11 117–11 128, https://doi.org/10.5194/acp-14-11117-2014, https://www.atmos-chem-phys.net/14/11117/2014/, 2014.

Radić, V. and Hock, R.: Modeling future glacier mass balance and volume changes using ERA-40 reanalysis and climate models: A sensitivity study at Storglaciären, Sweden~~: MODELINGSTORGLACIÄREN'SVOLUMEEVOLUTION~~, Journal of Geophysical Research: Earth Surface, 111, n/a–n/a, https://doi.org/10.1029/2005JF000440, http://doi.wiley.com/10.1029/2005JF000440, 2006.

Radić, V., Menounos, B., Shea, J., Fitzpatrick, N., Tessema, M. A., and Déry, S. J.: Evaluation of different methods to model near-surface turbulent fluxes for a mountain glacier in the Cariboo Mountains, BC, Canada, The Cryosphere, 11, 2897–2918, https://doi.org/10.5194/tc-11-2897-2017, https://www.the-cryosphere.net/11/2897/2017/, 2017.

Reijmer, C. H. and Hock, R.: Internal accumulation on Storglaciären, Sweden, in a multi-layer snow model coupled to a distributed energy- and mass-balance model, Journal of Glaciology, 54, 61–72, https://doi.org/https://doi.org/10.3189/002214308784409161, 2008.

Rye, C. J., Willis, I. C., Arnold, N. S., and Kohler, J.: On the need for automated multiobjective optimization and uncertainty estimation of glacier mass balance models~~: CALIBRATIONOFGLACIERMODELS~~, Journal of Geophysical Research: Earth Surface, 117, n/a–n/a, https://doi.org/10.1029/2011JF002184, http://doi.wiley.com/10.1029/2011JF002184, 2012.

Sauter, T. and Galos, S. P.: Effects of local advection on the spatial sensible heat flux variation on a mountain glacier, The Cryosphere, p. 19, https://doi.org/10.5194/tc-10-2887-2016, 2016.

Sauter, T. and Obleitner, F.: Assessing the uncertainty of glacier mass-balance simulations in the European Arctic based on variance decomposition, Geoscientific Model Development, 8, 3911–3928, https://doi.org/10.5194/gmd-8-3911-2015, https://www.geosci-model-dev.net/8/3911/2015/, 2015.

Sauter, T., Möller, M., Finkelnburg, R., Grabiec, M., Scherer, D., and Schneider, C.: Snowdrift modelling for the Vestfonna ice cap, north-eastern Svalbard, The Cryosphere, 7, 1287–1301, https://doi.org/10.5194/tc-7-1287-2013, https://www.the-cryosphere.net/7/1287/2013/, 2013.

Schuler, T. V., Hock, R., Jackson, M., Elvehøy, H., Braun, M., Brown, I., and Hagen, J.-O.: Distributed mass-balance and climate sensitivity modelling of Engabreen, Norway, Annals of Glaciology, 42, 395–401, https://doi.org/10.3189/172756405781812998, https://www.cambridge.org/core/product/identifier/S0260305500265324/type/journal_article, 2005.

Sicart, J. E., Hock, R., Ribstein, P., Litt, M., and Ramirez, E.: Analysis of seasonal variations in mass balance and meltwater discharge of the tropical Zongo Glacier by application of a distributed energy balance model, Journal of Geophysical Research, 116, D13 105, https://doi.org/10.1029/2010JD015105, http://doi.wiley.com/10.1029/2010JD015105, 2011.

Siemer, A. H.: Ein eindimensionales Energie-Massenbilanzmodell einer Schneedecke unter Berücksichtigung der Flüssigwassertransmission, Berichte des Institutes für Meteorologie und Klimatologie der Universität Hannover 34, Universität Hannover, Hannover, 1988.

Smeets, C. J. P. P. and van den Broeke, M. R.: Temporal and Spatial Variations of the Aerodynamic Roughness Length in the Ablation Zone of the Greenland Ice Sheet, Boundary-Layer Meteorology, 128, 315–338, https://doi.org/10.1007/s10546-008-9291-0, http://link.springer.com/10.1007/s10546-008-9291-0, 2008.

Stull, R. B., ed.: An Introduction to Boundary Layer Meteorology, Springer Netherlands, Dordrecht, https://doi.org/10.1007/978-94-009-3027-8, http://link.springer.com/10.1007/978-94-009-3027-8, 1988.

Van Den Broeke, M., Reijmer, C., Van As, D., and Boot, W.: Daily cycle of the surface energy balance in Antarctica and the influence of clouds, International Journal of Climatology, 26, 1587–1605, https://doi.org/10.1002/joc.1323, http://doi.wiley.com/10.1002/joc.1323, 2006.

van der Walt, S., Colbert, S. C., and Varoquaux, G.: The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22–30, https://doi.org/10.1109/MCSE.2011.37, http://ieeexplore.ieee.org/document/5725236/, 2011.

van Pelt, W. J. J., Oerlemans, J., Reijmer, C. H., Pohjola, V. A., Pettersson, R., and van Angelen, J. H.: Simulating melt, runoff and refreezing on Nordenskiöldbreen, Svalbard, using a coupled snow and energy balance model, The Cryosphere, 6, 641–659, https://doi.org/10.5194/tc-6-641-2012, https://www.the-cryosphere.net/6/641/2012/, 2012.

Vionnet, V., Brun, E., Morin, S., Boone, A., Faroux, S., Le Moigne, P., Martin, E., and Willemet, J.-M.: The detailed snowpack scheme Crocus and its implementation in SURFEX v7.2, Geoscientific Model Development, 5, 773–791, https://doi.org/10.5194/gmd-5-773-2012, https://www.geosci-model-dev.net/5/773/2012/, 2012.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, I., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and Contributors, S. . .: SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python, arXiv:1907.10121 [physics], http://arxiv.org/abs/1907.10121, arXiv: 1907.10121, 2019.

Wagnon, P., Ribstein, P., Kaser, G., and Berton, P.: Energy balance and runoff seasonality of a Bolivian glacier, Global and Planetary Change, 22, 49–58, https://doi.org/10.1016/S0921-8181(99)00025-9, https://linkinghub.elsevier.com/retrieve/pii/S0921818199000259, 1999.

Weidemann, S. S., Sauter, T., Malz, P., Jaña, R., Arigony-Neto, J., Casassa, G., and Schneider, C.: Glacier Mass Changes of Lake-Terminating Grey and Tyndall Glaciers at the Southern Patagonia Icefield Derived From Geodetic Observations and Energy and Mass Balance Modeling, Frontiers in Earth Science, 6, 81, https://doi.org/10.3389/feart.2018.00081, https://www.frontiersin.org/article/10.3389/feart.2018.00081/full, 2018.

Wever, N., Fierz, C., Mitterer, C., Hirashima, H., and Lehning, M.: Solving Richards Equation for snow improves snowpack meltwater runoff estimations in detailed multi-layer snowpack model, The Cryosphere, 8, 257–274, https://doi.org/10.5194/tc-8-257-2014, https://www.the-cryosphere.net/8/257/2014/, 2014.

Wohlfahrt, G., Hammerle, A., Niedrist, G., Scholz, K., Tomelleri, E., and Zhao, P.: On the energy balance closure and net radiation in complex terrain, Agricultural and Forest Meteorology, 226-227, 37–49, https://doi.org/10.1016/j.agrformet.2016.05.012, 2016.

Østby, T. I., Schuler, T. V., Hagen, J. O., Hock, R., Kohler, J., and Reijmer, C. H.: Diagnosing the decline in climatic mass balance of glaciers in Svalbard over 1957–2014, The Cryosphere, 11, 191–215, https://doi.org/10.5194/tc-11-191-2017, https://www.the-cryosphere.net/11/191/2017/, 2017.