MuSA: The ~~Multiscale~~ Multiple Snow Data Assimilation System (v1.0)

Esteban Alonso-González[1*], Kristoffer Aalstad[2*], Mohamed Wassim Baba[3], Jesús Revuelto[4], Juan Ignacio López-Moreno[4], Joel Fiddes[5], Richard Essery[6], and Simon Gascoin[1]

[1]Centre d'Etudes Spatiales de la Biosphère, Université de Toulouse,CNRS/CNES/IRD/INRA/UPS, Toulouse, France
[2]Department of Geosciences, University of Oslo, Oslo, Norway
[3]Center for Remote Sensing Application (CRSA), Mohammed VI Polytechnic University (UM6P), Ben Guerir, Morocco
[4]Instituto Pirenaico de Ecología, CSIC, Zaragoza, Spain
[5]WSL Institute for Snow and Avalanche Research SLF, Davos, Switzerland
[6]School of GeoSciences, University of Edinburgh, Edinburgh, UK
[*]These authors contributed equally to this work.

Correspondence: Esteban Alonso-González (esteban.alonso-gonzalez@univ-tlse3.fr)

Abstract. Accurate knowledge of the seasonal snow distribution is vital in several domains including ecology, water resources management, and tourism. Current spaceborne sensors provide a useful but incomplete description of the snowpack. Many studies suggest that the assimilation of remotely sensed products in physically based snowpack models is a promising path forward to estimate the spatial distribution of snow water equivalent (SWE). However, to date there is no standalone, open source ~~software~~ community-driven project dedicated to snow data assimilation which makes it difficult to compare existing algorithms, and fragments development efforts. Here we introduce a new data assimilation toolbox, the ~~Multiscale~~ Multiple Snow Data Assimilation System (MuSA), to help fill this gap. MuSA was developed to fuse remotely sensed information that is available at different timescales with the energy and mass balance Flexible Snow Model (FSM2). MuSA was designed to be user-friendly and scalable. It enables assimilation of different state variables such as the snow depth, SWE, snow surface temperature, binary or fractional snow-covered area, and snow albedo and could be easily upgraded to assimilate other variables such as liquid water content or snow density in the future. MuSA allows the joint assimilation of an arbitrary number of these variables, through the generation of an ensemble of FSM2 simulations. The characteristics of the ensemble (i.e. the number of particles and their prior covariance) may be controlled by the user, and it is generated by perturbing the meteorological forcing of FSM2. The observational variables may be assimilated using different algorithms including the particle filters and smoothers as well as ensemble Kalman filters and smoothers along with their iterative variants. We demonstrate the wide capabilities of MuSA through two snow data assimilation experiments. First, 5 m resolution snow depth maps derived from drone surveys are assimilated in a distributed fashion in the Izas catchment (Central Pyrenees). Furthermore, we conducted a joint assimilation experiment, fusing MODIS land surface temperature and fractional snow-covered area with FSM2 in a single cell experiment. In light of these experiments, we discuss the pros and cons of the assimilation algorithms, including their computational cost.

1 Introduction

The snow cover has a profound effect on the water cycle (García-Ruiz et al., 2011) and ecosystems (Lin and West, 2022) of high latitude and mountain regions. It represents a natural reservoir of freshwater resources, sustaining crop irrigation, hydropower generation and drinking water supply to a fifth of humanity (Barnett et al., 2005). In addition, the ski industry is an important economic driver in many mountain areas. As a consequence, a good knowledge of the snow cover properties has a strong scientific, societal, and economic value (Sturm et al., 2017).

Due to the harsh environmental conditions that often prevail in snow dominated areas, in situ monitoring of the snowpack based on automatic devices and weather stations is both costly and logistically challenging. In addition, due to the ~~variable nature~~ spatio-temporal variability of the snowpack (López-Moreno et al., 2011), even dense monitoring networks may suffer from a lack of representativeness ~~(Molotch and Bales, 2006)~~(Molotch and Bales, 2006; Cluzet et al., 2022). Yet, estimating SWE spatial distribution is important to make accurate predictions of snowmelt runoff in alpine catchments. Satellite remote sensing provides spatial information about snow-related variables including (i) snow cover spatial extent at various spatio-temporal scales (Aalstad et al., 2020; Gascoin et al., 2019; Hall et al., 2002; Hüsler et al., 2014), (ii) snow depth (Lievens et al., 2019; Marti et al., 2016), (iii) albedo (Kokhanovsky et al., 2020) or (iv) ~~snow~~ land surface temperature (Bhardwaj et al., 2017)~~at the global scale~~. However, the direct estimation of key variables such as the snow water equivalent (SWE) or density by means of remote sensing techniques, remains challenging (Dozier et al., 2016). The only remote sensing tools that have shown some potential to retrieve SWE are passive microwave sensors. Unfortunately, their coarse resolution and the fact that they tend to saturate above a certain SWE threshold, prevents their usage over mountainous regions or areas with a thick snowpack (Luojus et al., 2021).

Numerical ~~modeling allows the estimation~~ models can estimate the distribution of SWE at different spatio-temporal scales using meteorological information derived from automatic weather stations (Essery, 2015; Liston and Elder, 2006a) or atmospheric models (Alonso-González et al., 2018; Wrzesien et al., 2018). Nonetheless, such snowpack models exhibit a number of limitations that may cause strong biases in the SWE simulations (Wrzesien et al., 2017). Snowpack model uncertainties originate partly from their simplified representation of physical processes (Günther et al., 2019; Fayad and Gascoin, 2020), but most importantly from errors in the meteorological forcing (Raleigh et al., 2015). In this context, the fusion of remote sensing products with snowpack models using data assimilation is key to improve snowpack simulations (Girotto et al., 2020; Largeron et al., 2020).

Several remotely sensed products may be used to update snowpack models. ~~Snow covered area~~ Fractional snow-covered area (FSCA) from optical sensors was the first product to be assimilated (Clark et al., 2006; Durand et al., 2008; Kolberg and Gottschalk, 2006). ~~Snow cover area (SCA)~~ FSCA assimilation remains extensively used in both distributed (Margulis et al., 2016) and semi-distributed models (Thirel et al., 2013) due to the long time series of SCA observations and the development of new higher resolution products (Baba et al., 2018). It is possible to further improve snowpack simulations by assimilating snow depths (Deschamps-Berger et al., 2022; Smyth et al., 2020). The assimilation of remotely sensed surface reflectances may also be beneficial ~~(Charrois et al., 2016; Cluzet et al., 2020; ?)~~(Charrois et al., 2016; Cluzet et al., 2020; Revuelto et al., 2021b), but

further research is needed on this topic to demonstrate advantages over assimilating derived higher level products such as

55 ~~fractional snow cover area (FSCA)~~ FSCA and albedo.

Whereas snowpack models are increasingly available as open source software and remote sensing products as open data, to our knowledge there is no standalone open source application to develop multiple snow data assimilation experiments. This specific issue was highlighted by Fayad et al. (2017) as a strong limitation to advance knowledge on the snow cover in regions which receive less attention from the mainstream research community. In addition, some data assimilation frameworks

60 are based on highly specific implementations tied to operational constraints (Cluzet et al., 2021). This situation prevents the development of reproducible snow data assimilation studies and challenges the comparison of the performance of different snow data assimilation algorithms.

This is why we have developed a new open source data assimilation toolbox, the ~~Multiscale~~ Multiple Snow Data Assimilation system (MuSA). MuSA is an ensemble-based snow data assimilation tool. It enables the fusion of multiple observations with

65 a physically based snowpack model while taking into account various sources of ~~model, forcing ,~~ forcing and measurement uncertainty. It is an open-source collaborative project entirely written in the Python programming language. It should facilitate the development of snow data assimilation experiments, as well as the generation of snowpack reanalyses and near real time snowpack monitoring. It was designed with a modular structure to foster collaborative development, allowing advanced users to seamlessly implement new features with minimal effort. In the following sections, we describe the features of MuSA and

70 show different examples of its usage, assimilating remote sensing products of different spatio-temporal resolutions.

2 Overview of the data assimilation system

The core of MuSA is the energy and mass balance of the snowpack model, the Flexible Snow Model (FSM2 Essery, 2015). FSM2 has ~~from two to three levels of representation of different~~ several options for the parameterizations of key processes related to the energy and mass balance of the snowpack~~.~~. The most complex configuration is chosen by default in MuSA, leading

75 to a more detailed simulation of the internal snowpack processes. Albedo is computed from the age of the snow, decreasing its value as snow ages and increasing it with fresh snowfalls~~, instead of diagnosing it as a function of the surface temperature~~. Thermal conductivity of the snowpack is computed as a function of the snow density~~, instead of using a fixed value~~. Snow density is computed considering overburden and thermal metamorphism~~, instead of using an empirical estimation increasing the density as function of age or using a fixed value~~. Turbulent energy fluxes are computed as a function of atmospheric stability. Melt

80 water percolation in the snowpack is computed using the gravitational drainage~~, instead of a bucket model~~. Although this is the default configuration, it is possible to choose any other FSM2 setup, which may result in slight performance differences both in terms of the computational cost and accuracy of the model (Günther et al., 2019). As envisaged in Westermann et al. (2022), the potential to run multiple model configurations leaves the door open for pursuing rigorous model comparison using the model evidence framework (MacKay, 2003) in the cryospheric sciences.

85 MuSA was designed as a Python program encapsulating the FSM2 Fortran code (Figure 1). It handles the forcing and initial files as well as the FSM2 runs and outputs, by internally generating the ~~needed~~ required ensemble of simulations from

simple configuration (config.py) and constants (constants.py) files that should be filled by the user to both configure the MuSA environment and define the priors respectively. Then, it solves most of the challenges of ensemble-based snow data assimilation frameworks for the user, as it internally handles the I/O and parallelization while keeping track of the ensembles that should flow in different ways depending of the chosen DA algorithm. The data assimilation algorithms are independently implemented on a grid cell basis, allowing both ~~point scale~~ single cell and parallel spatially distributed simulations. ~~Also, in its current version MuSA provides support for the direct insertion of snow depth. However this feature may not be maintained in the future as it involves strong assumptions especially with multilayer geometries.~~ The outputs of MuSA consist of the posterior mean snow simulation from FSM2 (updated_idx_idy.csv), the posterior standard deviations of the FSM2 ensemble (sd_idx_idy.csv), ~~and~~ information related to the posterior perturbation parameters and the observations (DA_idx_idy.csv), and the original simulation without perturbation (OL_idx_idy.csv). Additionally, it is possible to store the posterior ~~ensembles~~ ensemble of every cell.
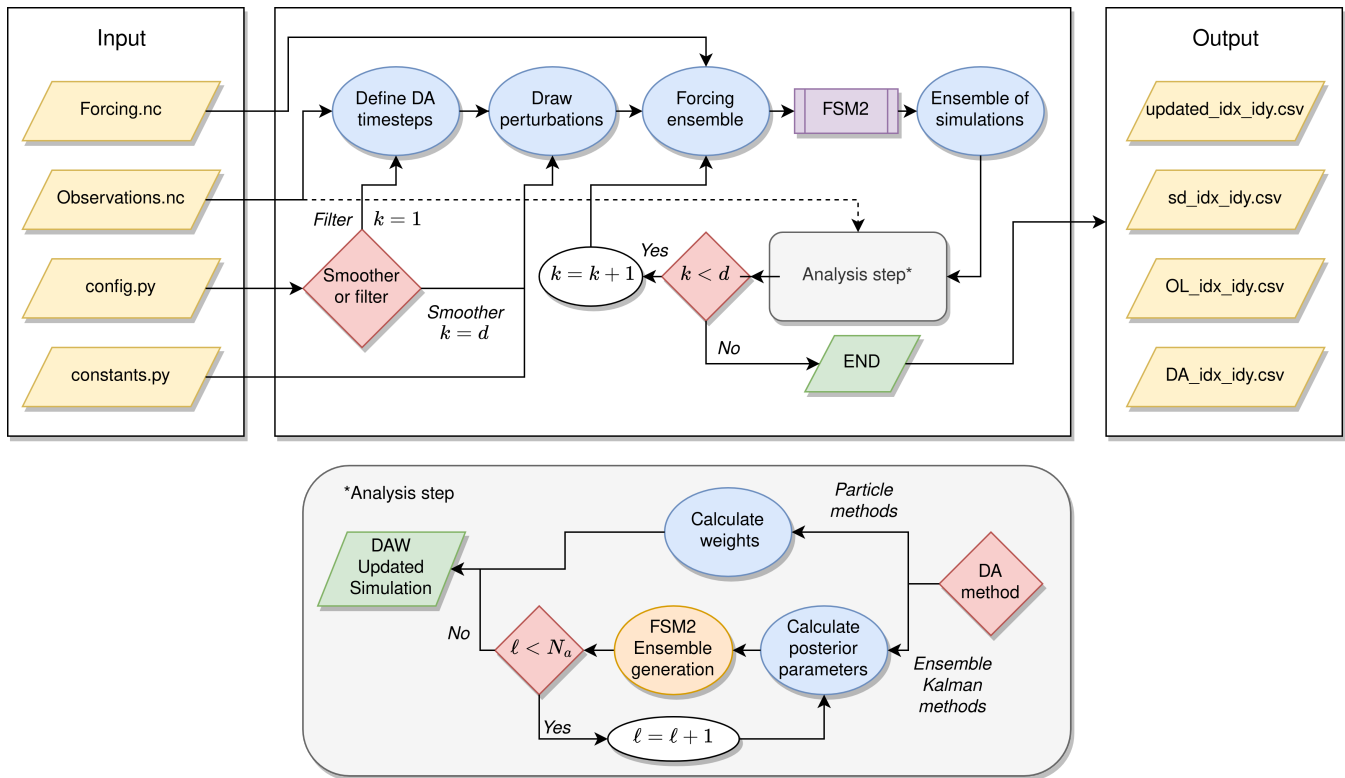
Figure 1. MuSA internal workflow. k, d, ℓ, N_a, DAW, idx, and idy refer to the timing of the analysis step, number of ~~observations~~ observation times, ~~Kalman iteration~~ current assimilation cycle number, total number of assimilation cycles, data assimilation window, longitude cell index, and latitude cell index, respectively.

The following data assimilation algorithms are currently implemented in MuSA:

– Particle Filter (PF; Gordon et al., 1993; van Leeuwen et al., 2019)

- Ensemble Kalman Filter (EnKF; Evensen, 1994; van Leeuwen, 2020)

- Ensemble Kalman filter with Multiple Data Assimilation (EnKF-MDA; Emerick and Reynolds, 2012)

- Particle Batch Smoother (PBS; Margulis et al., 2015), a batch smoother variant of the PF.

- Ensemble Smoother (ES; van Leeuwen and Evensen, 1996), a batch smoother variant of the EnKF.

- Ensemble Smoother with Multiple Data Assimilation (ES-MDA; Emerick and Reynolds, 2013)

In its current version, MuSA is able to assimilate the following variables:

- SWE (mm)

- Snow depth (m)

- Land surface temperature (K)

- Fractional snow-covered area (-)

- Albedo (-)

- Sensible heat flux to the atmosphere (W m^{-2})

- Latent heat flux to the atmosphere (W m^{-2})"

We expect to provide support for even more variables in the future. Note that the Ensemble Kalman schemes involving multiple data assimilation are iterative schemes ~~based on tempering the likelihood~~ (Stordal and Elsheikh, 2015; van Leeuwen et al., 2019). For the PF, several standard resampling algorithms (see Li et al., 2015, for a review) are available in MuSA, namely: bootstraping, residual resampling, stratified resampling, and systematic resampling. In addition to these standard resampling techniques, we have also implemented a heuristic approach based on redrawing from a normal approximation to the posterior which is loosely inspired by ~~the idea of using a proposal distributions~~ more advanced particle methods (Särkkä, 2013; van Leeuwen et al., 2019). This redrawing from the approximate posterior generates new samples of perturbation parameters at each assimilation step. In ~~the case of a total collapse of the PF with this resampling strategy the posterior standard deviation of the perturbation parameters is 0 (all weight is assigned to one particle) which corresponds to a degenerate distribution. In this case, drawing new perturbation parameters from the degenerate posterior distribution would not generate new particles. To avoid this , in case of a complete collapse of the PF, MuSA~~ case of complete degeneracy, where all the weight is on 1 particle, this redrawing technique instead uses the prescribed prior standard deviations for generating perturbation parameters, corrected by a scaling factor (to be selected by the user, fixed to 0.3 by default) to avoid ~~over-inflating~~ overdispersing the ensemble.

The inputs of MuSA are composed of (i) gridded meteorological forcing to generate the ensemble of FSM2 simulations and (ii) gridded observations which are assimilated. Optionally, it is possible to provide a mask, covering areas where simulations should not be implemented in order to save computational resources or restrict the runs to a specific area inside the provided

domain. These inputs must be provided as files encoded in the Network Common Data Form (NetCDF; Rew and Davis, 1990) format. MuSA is able to handle different observational products as long as the meteorological forcing and the observations

130 share the same geometry (i.e. same number of cells in the longitudinal and latitudinal dimensions). As in many other distributed snowpack models, there is currently no communication between grid cells in MuSA. This makes MuSA easily parallelizable as each cell is simulated and updated completely independently of the others (i.e., embarrassingly parallel). ~~As a result, the computational cost of the runs scales almost linearly with the number of computational units or the number of cells to solve respectively.~~ Different parallelization schemes are already implemented, including multiprocessing for single node runs and

135 message passing interface (MPI) and Portable Batch System job arrays, ~~providing~~ to provide support for computing clusters. Due to the embarrassingly parallel nature (no communication between tasks) of the MuSA computational problem, other parallelization schemes can easily be implemented. All the data processing is done by default in the temporary file system, but the user can choose any other location such as a specific temporary folder in a multi-node cluster or a random access memory drive in a local workstation to speed up the I/O processes.

140 ## 2.1 Ensemble generation

~~Other than for direct insertion, the~~ The DA algorithms implemented in MuSA all require a prior ensemble ~~(i.e. a collection)~~ of simulations to represent uncertainty. The number of ensemble members ~~(also known as particles; van Leeuwen et al., 2019) in the ensemble~~ (or, equivalently, particles; van Leeuwen, 2009), which we denote by N_e, should be specified by the user, as it drastically affects both the computational cost of the experiments and the performance of the data assimilation algorithms. To

145 generate an ensemble of snowpack state trajectories, MuSA perturbs the ~~forcing and performs~~ meteorological forcing to run an ensemble of FSM2 ~~runs with the perturbed forcing~~ simulations. An arbitrary number of forcing variables can be perturbed. The perturbation of the forcing is performed by drawing spatially independent random perturbation parameters from a normal distribution ~~defined by its standard deviation and mean, or the mean and standard deviation of the underlying normal function in case of choosing a~~ or log-normal ~~probability~~ distribution. The prior standard deviation and mean of ~~the distribution should~~

150 ~~be provided~~ these distributions should be specified by the user. MuSA supports normal or log-normal probability functions to generate ~~constant in time~~ additive or multiplicative perturbations respectively, depending on the physical bounds of the variable to be perturbed. ~~For example, additive perturbations can be used to generate air temperature~~ time series, whereas multiplicative Additive perturbations are typically used for air temperature. Multiplicative perturbations are recommended for precipitation to avoid negative values ~~, or~~ as well as for shortwave radiation to avoid negative values and positive night-time values.

155 Once the forcing is perturbed, MuSA partitions the precipitation between the liquid and solid phase. To estimate the phase of the precipitation, two different approaches are implemented. The simpler approach is based on a logistic function of the 2 m air temperature (Kavetski and Kuczera, 2007). A more complex psychrometric energy balance method is also implemented in MuSA. This approach uses the relative humidity and 2 m air temperature to infer the surface temperature of the falling hydrometeors, from which the phase of the precipitation can be estimated (Harder and Pomeroy, 2013).

2.2 Meteorological forcing

The variables required as forcing to run MuSA include downwelling (i.e., incoming) shortwave and longwave radiation $[\mathrm{W\,m^{-2}}]$, total (sum of liquid and solid) precipitation $[\mathrm{kg\,m^{-2}s^{-1}}]$, surface atmospheric pressure $[\mathrm{Pa}]$, 2 m air temperature $[\mathrm{K}]$, relative humidity $[\%]$, and 10 m wind speed $[\mathrm{m\,s^{-1}}]$. In the current version of MuSA, the forcing must be provided in the netCDF format at an hourly time step in a grid based geometry, without any specific requirement on the native spatial resolution. The most likely candidates to be used as meteorological forcing for MuSA will be the outputs of atmospheric simulations or automatic weather station data. These include atmospheric reanalyses, such as ERA5 (Hersbach et al., 2020) or MERRA-2 (Gelaro et al., 2017) and regional climate model outputs (Alonso-González et al., 2021) as well the outputs of different downscaling approaches driven by the aforementioned datasets (e.g. Fiddes and Gruber, 2014; Gutmann et al., 2016; Havens et al., 2017; Fiddes et al., 2022; Liston and Elder, 2006b). Other sources of information include automatic weather stations or gridded products derived from the interpolation of point scale meteorological information. The forcing variables should be provided in the International System of Units, but it is possible to perform simple unit transformations internally to facilitate the preprocessing of different forcing data sources, using the following affine transformation.

$$\psi_k = a\phi_k + b, \tag{1}$$

where ϕ_k (for $k = 0, \ldots, n$) is a forcing time series provided by the user with units $[\Phi]$ and ψ_k with converted units $[\Psi]$ is the transformed forcing time series after scaling by a with units $[\Psi\,\Phi^{-1}]$ and translating by b with units $[\Psi]$.

It should be noted that the forcing is read along the time dimension. As mentioned above, each Each grid cell is solved independently including which includes the reading of the forcing . However, this memory cost reduction comes with an associated computational cost due to the potential need of reading a long time dimension that occurs along the time dimension. Otherwise, each process would have to store considerably more data in memory leading to more costly I/O operations that would slow down the run time. Even so, just reading along the time dimension can come with a considerable computational cost if the time dimension is large. To alleviate this, the computational cost of MuSA time spent reading the forcing can be reduced by setting the chunk (a subset of the file to be read or written as a single I/O operation) of the netCDF forcing files along the time dimension, specially with long or large (i.e. many grid cells) simulations. To speed up the subsequent relaunching of the simulations when smoothing and filtering, MuSA generates an intermediate binary file with the forcing information of each cell needed to run a complete simulation . This accelerates the initialization of experiments that involve performing different MuSA runs for the same grid cells for each grid cell.

2.3 Observations and masked cells

MuSA is able to assimilate an arbitrary number of different observations that can be irregularly distributed in time and space. It is also possible to develop perform joint assimilation experiments that assimilate more than one variable type at the same time. To date, MuSA offers support to assimilate snow depth [m], snow water equivalent (SWE) [mm], surface temperature [K], snow-covered area (either binary or fractional) [-] and albedo [-]. A temporally and spatially constant scalar corresponding

to the assumed observation error variance must be provided for each ~~of the observed variables that are~~ type of observation that is to be assimilated. This assumption implies a diagonal observation error covariance matrix, $\mathbf{R}$, which is tantamount to assuming that observation errors are uncorrelated in both time and space. ~~The geometry of the observations must match the geometry of the forcing, i.e. both datasets must share the same grid extent and resolution~~Note that this formulation allows the user to account for differences in observation error that arise in the case when a variable is observed by multiple sensors with varying accuracy. By modifying the likelihood, it would also possible to account for non-Gaussian observation errors (Fletcher and Zupanski, 2006; Fowler and van Leeuwen, 2013), but this is not yet supported in MuSA. The time step ~~, however,~~ of the observations does not have to be the same as that of the forcing so that the observations can be irregularly spaced in time. Also, it is not necessary that the observations of the different variables occur at the same time step, allowing the assimilation of products from different satellite platforms with different orbits. Optionally, a binary mask can be provided in a separate file to delineate the basin of interest. If provided, MuSA will skip the cells that are not covered by the mask. Again, the mask must be provided on the same grid as the forcing and the observations. Many remotely sensed products exhibit gaps due to the presence of clouds or failures in the close sensing devices. As a general rule we do not recommend filling the gaps of the observed series. This filled data would exhibit a different error variance than the direct estimations from the remote sensors, introducing uncertainties which are not compatible with the data assimilation process. MuSA will internally handle the gaps in the observations, provided that the ~~netcdf~~ netCDF files are generated using the ~~regular~~ aforementioned conventions, assimilating just the timesteps where ~~the~~ information is available.

3 Data assimilation

Data assimilation (DA) is the exercise of fusing uncertain information from observations and (typically) geoscientific models (Evensen et al., 2022; Reich and Cotter, 2015). The DA schemes used in MuSA vary in their approach to state and parameter estimation as well as in the underlying assumptions and algorithms employed. In this section we provide an overview of these schemes as well as the underlying theory.

3.1 Bayesian inference

In line with most modern approaches to DA (Wikle and Berliner, 2007; Carrassi et al., 2018; van Leeuwen et al., 2019) the assimilation schemes used in MuSA are built on the foundation of Bayesian inference ~~(e.g. MacKay, 2003; Särkkä, 2013)~~ (e.g. Lindley, 2000; MacKay, 2003; Särkkä, 2013; Nearing et al., 2016). In this approach, the target quantity is the posterior distribution $p(\mathbf{x}|\mathbf{y})$, the probability of the model state $\mathbf{x} \in \mathbb{R}^m$ given the observations $\mathbf{y} \in \mathbb{R}^d$, which can be inferred using Bayes' rule

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})}, \tag{2}$$

in which $p(\mathbf{y}|\mathbf{x})$, the probability of the observed data given a model state, is called the likelihood while $p(\mathbf{x})$, the probability of the state before considering the observations, is known as the prior. According to the likelihood principle, the likelihood

should be seen as a function of the model state, not the fixed observations. So it is implicitly understood that the state is as a vector of random variables, while the observation vector is deterministic. In this Bayesian interpretation, rather than being narrowly synonymous with frequency, probability is a broad measure of uncertainty. The evidence (or marginal likelihood) in the denominator of (2) is given by

$$p(\mathbf{y}) = \int p(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{x} = \int p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \,, \tag{3}$$

where the range of integration is understood to be over all possible values of $\mathbf{x}$, namely the support of the prior. This evidence is just a function of the observations, ~~which are fixed, obtained by integrating the numerator (product of prior and the likelihood),~~ so it simply serves as normalizing constant in this incarnation of Bayes' theorem which we can ~~thus~~ rewrite as

$$p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}) \,, \tag{4}$$

to emphasize that the posterior is proportional to the product of the likelihood (what the data tells us) and the prior (what we ~~knew~~ believed before considering the data). ~~As~~ These are usually probability density rather than mass functions as we tend to deal with continuous variables in DA~~, the are usually probability density rather than mass functions~~. Since the posterior must integrate to one, we can safely ignore the evidence term as long as we are just dealing with the usual first level of inference where we fit a specific model. The evidence is nonetheless vital in the second level of inference where we compare models (see MacKay, 2003), but we do not pursue that here~~(see MacKay, 2003)~~. Thus, we will henceforth use the notation $Z = p(\mathbf{y})$ to emphasize that the evidence simply serves as a normalizing constant in this overview.

To get the Bayesian engine to run and infer the posterior, we need to specify the two input distributions on the right hand side of (2), namely the prior and the likelihood. To construct the likelihood, we specify an observation (or equivalently: measurement, data generating, forward) model of the form

$$\mathbf{y} = \mathcal{H}(\mathbf{x}^\star) + \boldsymbol{\epsilon} \,, \tag{5}$$

where $\mathbf{x}^\star$ contains the true values of the state variables, $\mathcal{H}(\cdot)$ is the observation operator which maps from state to observation space, and $\boldsymbol{\epsilon}$ is the observation error. ~~In practice~~ Often, as is the case in MuSA, the observation operator ~~essentially~~ picks out predicted observations, i.e. the state variables that correspond to observations, from the full state vector. As is commonly done in DA (Carrassi et al., 2018), we assume that the observation errors are additive, unbiased, and follow a Gaussian distribution, i.e. $\boldsymbol{\epsilon} \sim \mathrm{N}(\mathbf{0}, \mathbf{R})$ where $\mathbf{R}$ is the observation error covariance matrix. Thereby the likelihood becomes

$$p(\mathbf{y}|\mathbf{x}) = A \exp\left(-\frac{1}{2}\boldsymbol{\epsilon}^{\mathrm{T}} \mathbf{R}^{-1} \boldsymbol{\epsilon}\right) = A \exp\left(-\frac{1}{2}[\mathbf{y} - \widehat{\mathbf{y}}]^{\mathrm{T}} \mathbf{R}^{-1} [\mathbf{y} - \widehat{\mathbf{y}}]\right) \,, \tag{6}$$

where $A = \det(2\pi\mathbf{R})^{-1/2}$ is a normalizing constant, $(\cdot)^{\mathrm{T}}$ denotes the transpose, $\widehat{\mathbf{y}} = \mathcal{H}(\mathbf{x})$ are the predicted observations, and we have used that $\boldsymbol{\epsilon} = \mathbf{y} - \widehat{\mathbf{y}}$ conditional on $\mathbf{x}$ being true. Roughly speaking, the likelihood can be seen as a model misfit term which quantifies how likely the actual observations are given a particular model state. The states with a higher likelihood will correspond to those with predicted observations closer to the actual observations. The next ingredient is the prior distribution over states, $p(\mathbf{x})$, which can be specified based on initial beliefs which may include physical bounds, expert opinion, 'objective'

defaults (using e.g. maximum entropy), or knowledge from earlier analyses (Lindley, 2000; MacKay, 2003; Banner et al., 2020)

. For the latter, consider the case of assimilating two conditionally independent observations $\mathbf{y}_1$ and $\mathbf{y}_2$. Then the posterior can be obtained using either batch or recursive estimation (Särkkä, 2013). The batch approach involves solving directly for

$$p(\mathbf{x}|\mathbf{y}_1,\mathbf{y}_2) \propto p(\mathbf{y}_1,\mathbf{y}_2|\mathbf{x})p(\mathbf{x}) = p(\mathbf{y}_2|\mathbf{x})p(\mathbf{y}_1|\mathbf{x})p(\mathbf{x}), \tag{7}$$

while for the recursive approach we note that

$$p(\mathbf{x}|\mathbf{y}_1) \propto p(\mathbf{y}_1|\mathbf{x})p(\mathbf{x}), \tag{8}$$

so we can split up the batch solution by first conditioning on $\mathbf{y}_1$ to get an intermediate posterior, then use this as a prior when updating with $\mathbf{y}_2$ to get the full posterior as follows

$$p(\mathbf{x}|\mathbf{y}_1,\mathbf{y}_2) \propto p(\mathbf{y}_2|\mathbf{x})p(\mathbf{x}|\mathbf{y}_1). \tag{9}$$

This recursive approach will give the same answer as the batch approach, but it can often be helpful to split the inference into smaller chunks especially for dynamic problems. We can extend recursive estimation to an arbitrary number of observations as long as they are conditionally independent.

In theory, evaluating the posterior simply involves taking the product of two terms. Naïvely, this suggests that we can estimate the posterior through a simple grid approximation (MacKay, 2003). Unfortunately, inferring the posterior is usually akin to looking for a needle in a haystack due to the curse of dimensionality (Snyder et al., 2008) which manifests when we have a high dimensional state space and/or highly informative data. As such, we need to adopt efficient algorithms when doing DA in practice. Roughly speaking, the most widely used Bayesian DA schemes can be split into two major computational approaches: variational techniques and Monte Carlo methods (Carrassi et al., 2018). Despite success in operational meteorology (Bannister, 2017), the variational approach has received ~~little~~ less attention in cryospheric applications (Dumont et al., 2012) since it involves gradient terms that are difficult to implement and has not yet shown any clear gains in performance over Monte Carlo. As such, in MuSA we restrict ourselves to Monte Carlo methods where the naming reflects that these approaches rely on using (pseudo) random numbers to sample from probability distributions (MacKay, 2003).

3.2 Prediction, filtering, and smoothing

The subset of Monte Carlo methods used in MuSA can be split into two classes: those derived from the EnKF (Evensen, 1994) and the PF (Gordon et al., 1993). These can be further subdivided into filters and smoothers depending on how time dynamics are handled (Carrassi et al., 2018). As such, it is instructive to appreciate the differences between stochastic prediction, filtering, and smoothing (see Jazwinski, 1970; Särkkä, 2013) as well as how these are implemented in MuSA. To do so, we introduce the following notation: we consider discrete points in time $t_k = t_0 + k\Delta t$ for $k = 0, \ldots, n$ where t_0 is the initial time and Δt is the model timestep and we let $\mathbf{x}_k = \mathbf{x}(t_k)$ denote the state at a given time. We employ analogous notation for the observations $\mathbf{y}_k$, but (without loss of generality) we assume that the initial time is unobserved. Note that in practice we usually do not have observations at every model time step, we have just assumed this here to simplify notation. The extension of filters and

smoothers to the more realistic case with sparse observations in time is straightforward and merely requires a slight change in indexing so that filtering and smoothing only takes place using predicted observations at observation times.

In DA, prediction involves estimating the future state given past observations: $p(\mathbf{x}_k|\mathbf{y}_{1:\ell})$ where ~~$\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_\ell]$~~ $\mathbf{y}_{1:\ell} = [\mathbf{y}_1, \ldots, \mathbf{y}_\ell]$ and $\ell < k$. The archetypal example of this is the familiar task of weather forecasting through ensemble-based numerical weather prediction (Bauer et al., 2015). In the context of prediction, the concept of future and past are just relative to one another and not necessarily real time. So one can also do predictions of the past, as long as any observations considered are from the even more distant past. ~~Prediction also enters as~~ Since prediction is a step in both filtering and smoothing, ~~so it is helpful to understand~~ we explain how to implement it probabilistically. ~~The prediction step from timestep~~ Prediction from $k-1$ to k can be formulated as follows:

$$\mathbf{x}_k = \mathcal{M}(\mathbf{x}_{k-1}) + \boldsymbol{\eta}_{k-1}, \tag{10}$$

where $\mathcal{M}(\cdot)$ is the dynamical model (FSM2 in ~~this case~~MuSA) while $\boldsymbol{\eta} \sim \mathrm{N}(\mathbf{0}, \mathbf{Q})$ is the additive model error ~~(process noise)~~ term which we assume to be independent in time and follow a zero-mean Gaussian distribution with covariance matrix $\mathbf{Q}$. These assumptions can be relaxed without loss of generality, but their convenience and broad justifiability mean that they are often employed ~~in practice~~ (Carrassi et al., 2018). Crucially, the above prediction step produces Markovian (memoryless) dynamics where the current state depends only on the previous state and ~~white noise. This~~ noise. The Markov property is crucial in making the filtering and smoothing problems tractable. It implies $p(\mathbf{x}_k|\mathbf{x}_{0:(k-1)}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$ which lets us factorize and simplify distributions such as the full prior as follows

$$p(\mathbf{x}_{0:k}) = p(\mathbf{x}_k|\mathbf{x}_{0:(k-1)})p(\mathbf{x}_{0:(k-1)}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{0:(k-1)}), \tag{11}$$

where the transition density is Gaussian of the form $p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathrm{N}(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Q})$. Applying this recursively we obtain

$$p(\mathbf{x}_{0:k}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{x}_{k-2})\ldots p(\mathbf{x}_1|\mathbf{x}_0)p(\mathbf{x}_0) = p(\mathbf{x}_0)\prod_{j=1}^{k} p(\mathbf{x}_j|\mathbf{x}_{j-1}), \tag{12}$$

Using this kind of factorization together with marginalization also helps us construct the ~~(marginal)~~ marginal predictive distribution $p(\mathbf{x}_k|\mathbf{y}_{1:\ell})$ where $\ell < k$ as follows

$$p(\mathbf{x}_k|\mathbf{y}_{1:\ell}) = \int p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{y}_{1:\ell})\,\mathrm{d}\mathbf{x}_{k-1} = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:\ell})\,\mathrm{d}\mathbf{x}_{k-1}. \tag{13}$$

This is the Chapman-Kolmogorov equation (Särkkä, 2013) which can be applied recursively to obtain the predictive distribution at the current time step using the transition density together with previous predictive distributions.

From prediction we move to filtering which ~~involves estimating~~ is the estimation of the current state given current and past observations: $p(\mathbf{x}_k|\mathbf{y}_{1:k})$. This is the problem solved by sequential DA where an archetypal example is the initialization of numerical weather predictions as new observations become available to delay the effects of chaos (Bauer et al., 2015). To construct the filtering distribution we first re-introduce our Gaussian observation model in the dynamical context ~~where we~~ and make the usual assumption that the current observations are conditionally independent of both the observation and state

histories (Särkkä, 2013) resulting in the dynamic likelihood

$$p(\mathbf{y}_k|\mathbf{x}_{0:k}, \mathbf{y}_{1:(k-1)}) = p(\mathbf{y}_k|\mathbf{x}_k) = A_k \exp\left(-\frac{1}{2}\left[\mathbf{y}_k - \widehat{\mathbf{y}}_k\right]^{\mathrm{T}} \mathbf{R}_k^{-1}\left[\mathbf{y}_k - \widehat{\mathbf{y}}_k\right]\right), \tag{14}$$

where $\widehat{\mathbf{y}}_k = \mathcal{H}(\mathbf{x}_k)$ are the predicted observations ~~at the current timestep~~ and we have added a time index to the normalizing constant (A_k) and the observation error covariance matrix (~~$\mathbf{R}$), respectively,~~ $\mathbf{R}_k$) to emphasize that both the number and types of observations ~~at a given point in time may vary . Combining~~ may vary in time. By combining Markovian state dynamics with a conditionally independent observation model~~means that~~, we end up with a state-space or hidden Markov model (Cappé et al., 2005) where the states at each timestep are hidden (or latent) because they are not observable due to measurement error. The filtering distribution ~~can now be~~ is obtained by combining the predictive distribution ~~for $\ell = k - 1$~~ (which serves as the prior) and the dynamic likelihood through Bayes' theorem ~~as follows~~

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \propto p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:(k-1)}). \tag{15}$$

As such, prediction and filtering can be applied one after the other in time to sequentially obtain the filtering distributions of interest for all integration time-steps $k = 1, \ldots, n$. That is, starting from the initial prior $p(\mathbf{x}_0)$ we run the dynamical model to $k = 1$ to get the transition density in the product $p(\mathbf{x}_1|\mathbf{x}_0)p(\mathbf{x}_0)$ and marginalize to obtain the predictive distribution $p(\mathbf{x}_1)$ which we use as the prior when assimilating the observations $\mathbf{y}_1$ to estimate the filtering distribution $p(\mathbf{x}_1|\mathbf{y}_1)$. We continue this cycle, running the dynamical model to $k = 2$ to obtain $p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_1|\mathbf{y}_1)$ and marginalize to obtain $p(\mathbf{x}_2|\mathbf{y}_1)$ which we use as the prior when assimilating $\mathbf{y}_2$ to arrive at $p(\mathbf{x}_2|\mathbf{y}_{1:2})$. This filtering process of "online" assimilation as observations become available sequentially in time is appealing for operational forecasting since it can continue indefinitely with low memory requirements while outputting the filtering and prediction distributions of interest. In practice when using Monte Carlo methods, we do not operate with the distributions themselves but rather with an ensemble ~~(i.e. a collection)~~ of samples from these. Typically, due to the often prohibitively large size of a full spatio-temporal ensemble, one would only store summary statistics such as the posterior mean and standard deviation of the state for each point in space and time as the outputs of the analysis.

In addition to filtering, we are also able to solve some smoothing problems in MuSA. The key difference between filtering and smoothing is that the later considers future observations, i.e. the (marginal) smoothing distribution is ~~$p(\mathbf{x}_k|\mathbf{y}_{1:\ell}$~~ $p(\mathbf{x}_k|\mathbf{y}_{1:\ell})$ where $\ell > k$. Several types of smoothing problems exist (Cosme et al., 2012; Särkkä, 2013), namely: fixed-lag smoothing ($\ell = k + l$ with l constant) which is equivalent to filtering but where the state is lagged relative to the observations, fixed-point smoothing (k is fixed) where the posterior at a fixed point in time (such as an initial condition) is conditioned on both past and future observations, and fixed-interval smoothing (ℓ is fixed) where we estimate the posterior for points in a time interval given all observations in that interval. In MuSA we focus on a form of the fixed-interval smoothing known as batch smoothing in which the states within the time interval, henceforth called the data assimilation window (DAW), are updated in a single batch using all observations such that $\ell = n$.

The batch smoothing approach has been shown to be especially well suited for reanalysis type problems in land and snow data assimilation since it allows ~~for backward propagation of information~~ information from observations to propagate backwards

in time (Dunne and Entekhabi, 2005; Durand et al., 2008). We will restrict our attention to strong-constraint batch smoothing with indirect updates (Evensen, 2019; Evensen et al., 2022) where it is assumed that the dynamical model is perfect (i.e., no model error η) and that all uncertainty stems from parameters that are constant within the DAW. This approach is adopted both for simplicity as well as the fact that it ensures that the updated states are dynamically consistent with the physics in the model. It also has the advantage of implicitly ensuring that state variables such as snow depth or SWE remain within their physically non-negative bounds. In this approach, an ensemble of realizations of the model are run forward for an entire DAW during which all the observations and the corresponding predicted observations from the model are stored. At the end of the DAW, these observations are all assimilated simultaneously in a batch update to provide samples from the posterior parameter distribution. For seasonal snow the water year becomes a natural choice for the DAW. With this choice, potentially highly informative observations made during the ablation season are able to inform the preceding accumulation season (Margulis et al., 2015). This ~~backwards propagation of information is crucial to help reconstruct peak SWE which~~ is ~~usually of particular~~ crucial in peak SWE reconstruction which is of great interest to snow hydrologists (Dozier et al., 2016).

3.3 Consistency

A potential problem in DA is that the stochastic perturbations and updates that are imposed on the model and the resulting dynamics may lead to inconsistent model states. We can define (at least) two different types of inconsistencies: (i) dynamical inconsistency and (ii) physical inconsistency where model states and/or parameters violate their physical bounds.

Dynamical inconsistency with respect to the underlying deterministic model can enter into the DA exercise either through stochastic model error terms in prediction steps or through the assimilation step itself. In the case of weak constraint DA, where imperfections in the model are represented by stochastic model error terms, this kind of weak inconsistency exists by design and is a feature rather than a bug. In particular, the dynamical inconsistency created by the stochastic terms is meant to explicitly account for model uncertainty arising for example due to unresolved processes (Palmer, 2019). As such, the open-loop (no DA) stochastic model dynamics will be inconsistent with a deterministic version of the model. Arguably calling such a stochastic model dynamically inconsistent is somewhat of a misnomer, since it aspires to be more consistent with reality than a purely deterministic version of the same model.

The assimilation step itself can also introduce dynamical inconsistencies, particularly when applying filtering algorithms (Dunne and Entekhabi, 2005). For the EnKF these manifest as sawtooth-like patterns (jumps) in the dynamics at each assimilation step due to the fact that the Kalman-based analysis actually moves the ensemble from being samples of the prior to being (approximate) samples from the posterior. For the PF dynamical inconsistency also enters at the assimilation step whenever resampling is performed. The resampling step tends to kill off unpromising low weight particles while reproducing more promising higher weight particles. Since this evolutionary step is implemented sequentially in time when filtering, it will also introduce discontinuities in the dynamics of the model ensemble. In the case of both the EnKF and the PF, the discontinuity introduced at the assimilation step is an expected result of filtering. If one wishes to avoid this behavior, one should instead turn to smoothing algorithms.

Physical inconsistency can also enter both through stochastic perturbations and the assimilation step itself. Many snowpack state variables (and potential observables) have a relatively limited dynamic range and are either low bounded, e.g. snow depth and SWE are non-negative, or double bounded, e.g. snow albedo and FSCA are physically confined to the range $[0, 1]$. This means that directly applying assimilation routines which assume unbounded (e.g. Gaussian) distributions on the prior, likelihood, or the model error can lead to physical inconsistencies in snow DA.

A simple but naive solution is to just explicitly enforce state variables to lie within their physical bounds using minimum and/or maximum functions. Unfortunately, this is the same as truncating the underlying probability distributions and may degrade the performance of the assimilation scheme due to a strong violation of assumptions. Although this is a major concern for the EnKF-based schemes, which make explicit assumptions about Gaussianity, in practice it may also impact the PF where Gaussian distributions remain a convenient choice. A better way to deal with issues around physical inconsistency is to apply transformations to bounded random variables. Typically, these transformations will map such random variables from the bounded physical space to an unbounded space that can accommodate Gaussian distributions through Gaussian anamorphosis techniques (Bertino et al., 2003)). This can be carried out using transforms that have been empirically constructed or using analytic functions. In MuSA, following ~~(Aalstad et al., 2018)~~ Aalstad et al. (2018), we employ analytical Gaussian anamorphosis where we use a logarithm transform for variables that are physically lower bounded and a logit transform for variables that are physically double bounded. The corresponding inverse functions, i.e. the exponential and logistic transforms, can be used to map these unbounded variables back to the bounded physical space. For bounded variables we will often construct the prior distributions using these transforms. For example, assigning a lognormal prior can ensure that a variable never exceeds its lower bounds. In such a case the assimilation step takes place in the unbounded log transformed space where this variable is normally (i.e. Gaussian) distributed while the model integration takes place in the bounded physical space, which can be recovered using the exponential transform, where the variable is lognormally (i.e. log-Gaussian) distributed.

A key step taken in MuSA to reduce inconsistencies in snow data assimilation is to split the state vector in two: ~~$\mathbf{x} = [\mathbf{uv}]$~~ $\mathbf{x} = [\mathbf{u}, \mathbf{v}]$ where $\mathbf{u} \in \mathbb{R}^{m_p}$ are parameters while $\mathbf{v} \in \mathbb{R}^{m_s}$ are internal model states. Only the parameters, which can include both internal model parameters and forcing perturbation parameters, are assumed to be random variables in MuSA. This ~~appraoch~~ approach corresponds to the so-called forcing formulation (as opposed to the model-state formulation) of the DA problem (Evensen et al., 2022). The internal model states are taken to be deterministic given these parameters. Such a setup is tantamount to assuming that all the uncertainty in the model dynamics stems from the forcing data and internal parameters. This assumption is largely in line with previous findings (Raleigh et al., 2015) particularly for intermediate complexity snow models such as FSM2 (Günther et al., 2019). This is gradually becoming a relatively standard setup in snow data assimilation experiments ~~(Margulis et al., 2015; Magnusson et al., 2017; Aalstad et al., 2018; Alonso-González et al., 2021)~~ (Margulis et al., 2015; Magnusso. although it has arguably not been as properly formalized as elsewhere in the DA literature (Evensen, 2019; Evensen et al., 2022). Through such a split we ensure that internal model states remain both dynamically and physically consistent given the parameters. The parameters, in turn, are made physically consistent by applying analytical Gaussian anamorphosis transforms in the DA scheme. As such, we will let $\mathbf{u}$ denote anamorphosed parameters that have undergone a forward transform to the

unbounded space. It is implicitly assumed that these are inverse transformed to the physical space before they are passed to ~~the model~~ FSM2 to help evolve the internal states forward in time.

In the filtering algorithms in MuSA, both the stochastic parameters and the conditionally deterministic internal states are dynamic. The parameter dynamics evolve according to simple jitter (Farchi and Bocquet, 2018) of the form:

$$\mathbf{u}_k = \mathbf{u}_{k-1} + \boldsymbol{\eta}_{k-1}, \tag{16}$$

where we recall that $\boldsymbol{\eta}_{k-1} \sim \mathrm{N}(\mathbf{0}, \mathbf{Q})$ and that the parameters are defined in the unbounded transformed space. Probabilistically, this corresponds to the Markovian transition density $p(\mathbf{u}_k|\mathbf{u}_{k-1}) = \mathrm{N}(\mathbf{u}_k|\mathbf{u}_{k-1}, \mathbf{Q})$. The internal state variables, on the other hand, evolve according to the full dynamical model (FSM2) which also depends on the dynamic parameters $\mathbf{u}_k$ through $\mathbf{v}_k = \mathcal{M}(\mathbf{u}_k, \mathbf{v}_{k-1})$. ~~We reiterate that it is implicitly understood that these parameters are transformed back to physical space when they are used in the dynamical model.~~ These dynamics corresponds to the transition density (e.g. Evensen, 2018)

$$p(\mathbf{v}_k|\mathbf{u}_k, \mathbf{v}_{k-1}) = \delta(\mathbf{v}_k - \mathcal{M}(\mathbf{u}_k, \mathbf{v}_{k-1})), \tag{17}$$

where $\delta(\cdot)$ is the Dirac delta function which emphasizes that the internal state is not only Markovian but also conditionally deterministic given the parameters. This just formalizes that the only uncertainty in the internal state dynamics stems from the parameters to which we assign an initial prior $p(\mathbf{u}_0)$. ~~When~~ Without loss of generality and to simplify the presentation of the theory, we assume that we are dealing with seasonal snow, so that the initial prior for the internal snow states, $p(\mathbf{v}_0)$, is known $\mathbf{v}_0 = \boldsymbol{\nu}_0$ ~~and thus deterministic so~~. Thus $p(\mathbf{v}_0) = \delta(\mathbf{v}_0 - \boldsymbol{\nu}_0)$ since the annual integration period starts at the beginning of the water year where we assume that a snowpack has not yet formed so internal snow state variables are either 0 (snow depth, SWE) or undefined (e.g. snow surface temperature). Together with the dynamic likelihood $p(\mathbf{y}_k|\mathbf{u}_k)$, these distributions can be used to construct the target marginal filtering distribution $p(\mathbf{u}_k|\mathbf{y}_{1:k})$ which we can estimate with ensemble Kalman or particle filtering algorithms. By combining this marginal with appropriate densities we can also estimate the joint filtering distribution $p(\mathbf{u}_k, \mathbf{v}_k|\mathbf{y}_{1:k})$ from which we can compute posterior expectations for the internal states of interest. The derivation of this distribution is relatively analogous to that for the smoother below, but involves more steps and is thus not included herein. In practice, (approximate) samples from this joint filtering distribution are obtained from the posterior ensemble after each observation time and the subsequent assimilation step while running the filtering schemes in MuSA.

For the smoothing algorithms in MuSA, we instead assume that the dynamical model is perfect so the components of $\mathbf{u}$ are constant (i.e. time-invariant) but uncertain parameters, which can be either internal parameters or related to the forcing, while $\mathbf{v}_{0:n} = [\mathbf{v}_0, \ldots, \mathbf{v}_k, \ldots, \mathbf{v}_n]$ remain the dynamic internal state variables with initial state $\mathbf{v}_0$ at t_0 and final state $\mathbf{v}_n$ at t_n which is the end of what is now an annual DAW. The prediction step with the assumed perfect dynamical model then becomes $\mathbf{v}_k = \mathcal{M}(\mathbf{u}, \mathbf{v}_{k-1})$ which implies the following transition density:

$$p(\mathbf{v}_k|\mathbf{v}_{k-1}, \mathbf{u}) = \delta(\mathbf{v}_k - \mathcal{M}(\mathbf{u}, \mathbf{v}_{k-1})). \tag{18}$$

This is just a formal way of denoting that the only uncertainty in the dynamics stems from time invariant but uncertain parameters to which we assign a prior. ~~Thereby~~ As noted by Evensen (2019), when these perturbation parameters are interpreted

as errors this can be viewed as a limiting case of perfect time-correlation where the errors become constant biases. The lack of dynamics in the perturbation parameters is thus akin to assuming that errors in the forcing are constant in time within a particular water year. This simplifying assumption imposes a longer memory in the system than with jitter and facilitates the propagation of information backwards in time using smoothers. Given this transition density, the full prior for the entire DAW can be factorized as follows

$$p(\mathbf{x}) = p(\mathbf{u}, \mathbf{v}_{0:n}) = p(\mathbf{v}_{0:n}|\mathbf{u})p(\mathbf{u}) = p(\mathbf{u})p(\mathbf{v}_0) \prod_{k=1}^{n} p(\mathbf{v}_k|\mathbf{v}_{k-1}, \mathbf{u}), \tag{19}$$

where as before $p(\mathbf{v}_0) = \delta(\mathbf{v}_0 - \boldsymbol{\nu}_0)$ is the prior for the initial condition of the state. The full posterior then becomes

$$p(\mathbf{x}|\mathbf{y}_{1:n}) = p(\mathbf{u}, \mathbf{v}_{0:n}|\mathbf{y}_{1:n}) \propto p(\mathbf{y}_{1:n}|\mathbf{u})p(\mathbf{u}, \mathbf{v}_{0:n}), \tag{20}$$

which can be marginalized to obtain the marginal posterior for the parameters

$$p(\mathbf{u}|\mathbf{y}_{1:n}) \propto \int p(\mathbf{y}_{1:n}|\mathbf{u})p(\mathbf{u}, \mathbf{v}_{0:n}) \, \mathrm{d}\mathbf{v}_{0:n} = p(\mathbf{y}_{1:n}|\mathbf{u})p(\mathbf{u}), \tag{21}$$

where the batch likelihood is

$$p(\mathbf{y}_{1:n}|\mathbf{u}) = A_{1:n}\exp\left(-\frac{1}{2}[\mathbf{y}_{1:n} - \widehat{\mathbf{y}}_{1:n}]^{\mathrm{T}}\mathbf{R}_{1:n}^{-1}[\mathbf{y}_{1:n} - \widehat{\mathbf{y}}_{1:n}]\right), \tag{22}$$

in which $\widehat{\mathbf{y}}_{1:n} = [\widehat{\mathbf{y}}_1, \ldots, \widehat{\mathbf{y}}_k, \ldots, \widehat{\mathbf{y}}_n]$ contains the predicted observations $\widehat{\mathbf{y}}_k = \mathcal{H}(\mathbf{u}, \mathbf{v}_k)$ for all observation time steps, while $A_{1:n} = \det(2\pi\mathbf{R}_{1:n})^{-1/2}$ where $\mathbf{R}_{1:n} = \mathrm{diag}(\mathbf{R}_1, \ldots, \mathbf{R}_k, \ldots, \mathbf{R}_n)$ is the batch observation error covariance matrix which is a block diagonal matrix containing the observation error covariance matrices for all observation time steps. Note since that the state $\mathbf{v}_{0:n}$ is conditionally deterministic given $\mathbf{u}$, it is not explicitly included in the batch likelihood. By combining the batch likelihood with the prior for the parameters, we have all that we need to estimate the posterior for the parameters $p(\mathbf{u}|\mathbf{y}_{1:n}) \propto p(\mathbf{y}_{1:n}|\mathbf{u})p(\mathbf{u})$. The main difference to the filtering distribution is that all the data are assimilated in a single batch to update static parameters. Once we have obtained the posterior for the parameters, due to the strong constraint, we recall that we can easily recover the joint posterior for the states and parameters $p(\mathbf{v}_{0:n}, \mathbf{u}|\mathbf{y}_{1:n}) \propto p(\mathbf{v}_{0:n}|\mathbf{u})p(\mathbf{u}|\mathbf{y}_{1:n})$ from which we can compute posterior expectations for the internal states of interest. In practice, the steps involved in estimating the posterior and subsequent expectations depend on which batch smoothing algorithm is used.

3.4 Particle methods

Particle methods (Särkkä, 2013)(van Leeuwen, 2009; Särkkä, 2013), also known as Sequential Monte Carlo (Cappé et al., 2005; Chopin and Papaspiliopoulos, 2020), are appealing since they impose few assumptions, can be relatively easy to derive, and are simple to implement in their basic form. In principle, they can be used to solve any Bayesian inference problem, including the aforementioned filtering and smoothing problems. The so-called bootstrap filter described by Gordon et al. (1993) arguably marks the introduction of particle filters to the wider scientific community. Nonetheless, the underlying idea of sequential importance resampling (SIR) was already well known in statistics literature from which it emerged (see Smith and

Gelfand, 1992, and references therein). For a more up to date perspective, van Leeuwen et al. (2019) provide a comprehensive review of particle methods for geoscientific data assimilation, including a discussion of the state-of-the-art and promising avenues for further developments.

Particle methods have ~~rapid~~ rapidly become the most widely adopted approach to data assimilation within the snow science community. The PF, in particular, has become increasingly popular in snow data assimilation studies. This began with the seminal work of Leisenring and Moradkhani (2011) and Dechant and Moradkhani (2011) showing how the PF can outperform the EnKF in snow data assimilation applications, assimilating SNOTEL SWE and passive microwave data, at least if relatively simple models and a large ensemble is used. Charrois et al. (2016) subsequently applied the PF in synthetic (twin) experiments that assimilated MODIS-like reflectance data into the Crocus snow model for a site in the French Alps. Magnusson et al. (2017) showed how snow depth assimilation using the PF improved the estimation of several variables, including SWE and snowmelt runoff, in an intermediate complexity snow model for many sites across the Alps. The study of Baba et al. (2018) applied the PF to assimilate novel SCA satellite retrievals from Sentinel-2, demonstrating marked improvements in snowpack simulations across the sparsely instrumented High-Atlas mountains of Morocco. Piazzi et al. (2018) investigated the potential of performing a joint ~~(i.e., multivariate)~~ assimilation of several snowpack variables obtained from ground-based observations at 3 sites in the Alps using the PF, noting that degeneracy was more likely to occur for this setup than in more typical uni-variate snow data assimilation experiments. Smyth et al. (2019) assimilated monthly snow depth observations into an intermediate complexity snow model with the PF at a well instrumented site in the California Sierra Nevada, improving both snow density and SWE estimates with promising implications for the assimilation of snapshots of snow depth retrieved from airborne and space-borne platforms. These results were confirmed by Deschamps-Berger et al. (2022) who showed that the assimilation of a single snow depth map per season is sufficient to improve the simulated spatial variability of the snowpack. Recent efforts ~~, such as the synthetic study of Cluzet et al. (2021),~~ (Cantet et al., 2019; Cluzet et al., 2021, 2022; Odry et al., 2022) have focused on the challenge of spatially propagating snowpack information from observed to unobserved locations when assimilating in-situ snow depth and SWE data using the PF.

Another line of studies has adopted particle smoothing schemes, particularly the particle batch smoother (PBS Margulis et al., 2015), for snow reanalysis. Here a batch of remotely sensed, typically FSCA, observations are assimilated simultaneously to weight (assign probability mass to) an ensemble of model trajectories for the entire water year. Margulis et al. (2016) used the PBS to perform a 90 meter resolution snow reanalysis for the California Sierra Nevada covering the entire Landsat-era from 1985 to 2015. Cortés and Margulis (2017) applied a similar setup to conduct a snow reanalysis for the extratropical Andes. Aalstad et al. (2018) compared the performance of the PBS with ensemble Kalman-based smoothers when assimilating FSCA data from Sentinel-2 and MODIS for sites in the high-Arctic and found that the PBS markedly outperformed ~~non-iteratve~~ non-iterative ensemble Kalman-based smoothers in line with Margulis et al. (2015). Baldo and Margulis (2018) developed a multi-resolution snow reanalysis framework using the PBS and demonstrated, through tests in a basin in Colorado, that this could match the performance of the original single-resolution approach at a fraction of the computational cost. In DA experiments in the Swiss Alps, Fiddes et al. (2019) showed how clustering offers an alternative promising avenue for speeding up snow DA with the PBS, allowing for hyper-resolution reanalyses. Alonso-González et al. (2018) carried out

a snow reanalysis over the sparsely instrumented Lebanese mountains by forcing FSM with quasi-dynamically downscaled meteorological reanalysis data and subsequently assimilating MODIS-based FSCA retrievals with the PBS. Liu et al. (2021) recently performed an 18-year 500 m resolution snow reanalysis for High Mountain Asia by using the PBS to jointly assimilate MODIS and Landsat FSCA data. Although all of these studies have focused on using FSCA data for snow reanalysis, other remotely sensed retrievals could also be considered. For example, the work of Margulis et al. (2019) has shown that the PBS can also be used to assimilate infrequent lidar-based snow depth retrievals with marked improvements in the estimation of snowpack-related variables. ~~The next logical step would thus be to jointly assimilate FSCA and emerging satellite-based snow depth retrievals Marti et al. (2016); Treichler and Kääb (2017); Lievens et al. (2019).~~

Importance sampling is key to understanding these particle methods. Despite what the name may suggest, this is actually not an approach to directly generate samples from a target distribution of interest. Instead, it allows us to estimate expectations of functions with respect to a target distribution by drawing from another distribution that it is easier to sample from (MacKay, 2003). In DA, the posterior is the target distribution of interest and the expectation of some function $g(\mathbf{x})$ with respect to the posterior is defined as follows (Särkkä, 2013)

$$\mathrm{E}[g(\mathbf{x})|\mathbf{y}] = \int g(\mathbf{x})p(\mathbf{x}|\mathbf{y})\,\mathrm{d}\mathbf{x}, \tag{23}$$

where the expectation of $g(\mathbf{x}) = \mathbf{x}$ yields the posterior mean $\widehat{\boldsymbol{\mu}}$ while the expectation of $g(\mathbf{x}) = (\mathbf{x} - \widehat{\boldsymbol{\mu}})^2$ yields the posterior variance $\sigma^2 \widehat{\sigma}^2$. If we could generate N N_e independent samples from the posterior, $\mathbf{x}^{(i)} \sim p(\mathbf{x}|\mathbf{y})$, we could estimate the expectation in (23) numerically using direct Monte Carlo integration as follows

$$\mathrm{E}[g(\mathbf{x})|\mathbf{y}] \simeq \frac{1}{N}\frac{1}{N_e}\sum_{i=1}^{N\ N_e} g\left(\mathbf{x}^{(i)}\right), \tag{24}$$

where the $\mathbf{x}^{(i)}$ with $i = 1, \ldots, N$ denote N $i = 1, \ldots, N_e$ denote N_e independent samples from the posterior. Thanks to the law of large numbers and the central limit theorem, we know that this approximation will converge almost surely to the true expectation as $N \to \infty$ $N_e \to \infty$ with a standard error inversely proportional to $\sqrt{N}$ $\sqrt{N_e}$ (Chopin and Papaspiliopoulos, 2020). Unfortunately, we are rarely able to generate independent samples directly from the posterior.

In importance sampling, we instead use a proposal (or importance) distribution $q(\mathbf{x})$ that we know how to sample from which must have at least the same support as the posterior (i.e. $q(\mathbf{x}) > 0$ wherever $p(\mathbf{x}|\mathbf{y}) > 0$). Then, by multiplying the integrand with $1 = q(\mathbf{x})/q(\mathbf{x})$, we can solve (23) using importance sampling-based Monte Carlo integration

$$\mathrm{E}[g(\mathbf{x})|\mathbf{y}] = \int g(\mathbf{x})\frac{p(\mathbf{x}|\mathbf{y})}{q(\mathbf{x})}q(\mathbf{x})\,\mathrm{d}\mathbf{x} \simeq \frac{1}{N}\frac{1}{N_e}\sum_{i=1}^{N\ N_e} g(\mathbf{x}^{(i)})\frac{p(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}\frac{p(\mathbf{x}^{(i)}|\mathbf{y})}{q(\mathbf{x}^{(i)})} = \frac{1}{N}\frac{1}{N_e}\sum_{i=1}^{N\ N_e} g(\mathbf{x}^{(i)})\widehat{w}(\mathbf{x}^{(i)}), \tag{25}$$

where the $\mathbf{x}^{(i)}$ are now samples from the proposal and we have defined the normalized weights $\widehat{w}(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})/q(\mathbf{x})$. Following the nomenclature of Chopin and Papaspiliopoulos (2020), these weights are normalized in the sense that their expectation with respect to $q(\mathbf{x})$ equals 1. A hurdle remains in that we only know the posterior up to an unknown normalizing constant (i.e., the evidence Z) so in practice we can only evaluate the un-normalized posterior. Recalling the definition of the evidence in (3), we

can nonetheless use importance sampling to approximate it as follows

$$Z = \int f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int \frac{f(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})\,\mathrm{d}\mathbf{x} \simeq \frac{1}{\underline{N}\,\underset{\sim}{N_e}}\frac{1}{}\sum_{i=1}^{\underline{N}\,\underset{\sim}{N_e}}\widetilde{w}(\mathbf{x}^{(i)}), \tag{26}$$

where $f(\mathbf{x}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$ is the un-normalized posterior and we have defined the un-normalized weights $\widetilde{w}(\mathbf{x}) = f(\mathbf{x})/q(\mathbf{x})$ which no longer have an expectation of 1 with respect to $q(\mathbf{x})$. Using the samples $\mathbf{x}^{(i)} \sim q(\mathbf{x})$, this evidence approximation, and that $p(\mathbf{x}|\mathbf{y}) = f(\mathbf{x})/Z$, we can now solve (23) as follows

$$\mathrm{E}[g(\mathbf{x})|\mathbf{y}] = \frac{1}{Z}\int \frac{f(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})\,\mathrm{d}\mathbf{x} \simeq \sum_{i=1}^{\underline{N}\,\underset{\sim}{N_e}}g(\mathbf{x}^{(i)})w(\mathbf{x}^{(i)}), \tag{27}$$

where, letting $w^{(i)} = w(\mathbf{x}^{(i)})$ for economy, the *auto*-normalized weights are given by $\cancel{w^{(i)} = \widetilde{w}^{(i)}\left(\sum_{j=1}^{N}\widetilde{w}^{(j)}\right)^{-1}}\,\underset{\sim}{w^{(i)} = \widetilde{w}^{(i)}\left(\sum_{j=1}^{N_e}\widetilde{w}^{(j)}\right)}$ with the property that $\cancel{\sum_{i=1}^{N}w^{(i)} = 1}\,\underset{\sim}{\sum_{i=1}^{N_e}w^{(i)} = 1}$. Note that with these auto-normalized weights $q(\mathbf{x})$ also only needs to be known up to a normalizing constant Z_q since any such constants cancel out in the auto-normalization step. Furthermore, direct Monte Carlo integration can be seen as a special case of importance sampling where the proposal is the target distribution itself which leads to uniformly equal weights with a value of $\cancel{1/N}\,\underset{\sim}{1/N_e}$.

Mathematically, importance sampling is tantamount to a particle representation of the posterior through a sum of weighted Dirac delta functions centered on the sampled states $\mathbf{x}^{(i)} \sim q(\mathbf{x})$ of the form $\cancel{p(\mathbf{x}|\mathbf{y}) \simeq \sum_{i=1}^{N}w^{(i)}\delta(\mathbf{x} - \mathbf{x}^{(i)})}\,\underset{\sim}{p(\mathbf{x}|\mathbf{y}) \simeq \sum_{i=1}^{N_e}w^{(i)}\delta(\mathbf{x} - \mathbf{x}^{(i)})}$ (Särkkä, 2013). We can see this by recalling that the Dirac delta has the properties that $\int \delta(\mathbf{x} - \mathbf{x}^{(i)})\,\mathrm{d}\mathbf{x} = 1$ and $\int g(\mathbf{x})\delta(\mathbf{x} - \mathbf{x}^{(i)})\,\mathrm{d}\mathbf{x} = g\left(\mathbf{x}^{(i)}\right)$, such that by inserting the particle representation in (23) we have

$$\mathrm{E}[g(\mathbf{x})|\mathbf{y}] \simeq \int \sum_{i=1}^{\underline{N}\,\underset{\sim}{N_e}}g(\mathbf{x})w^{(i)}\delta\left(\mathbf{x} - \mathbf{x}^{(i)}\right)\,\mathrm{d}\mathbf{x} = \sum_{i=1}^{\underline{N}\,\underset{\sim}{N_e}}g(\mathbf{x}^{(i)})w^{(i)}, \tag{28}$$

which is equal to the result in (27). This particle representation is helpful since we can conceptualize distributions as consisting of a set of particles (or points) in state space whose probability mass are given by their weights.

The catch with importance sampling is that, unless the proposal is nearly identical to the target distribution, all the probability mass tends to collapse onto just a few particles as the dimensions of a problem increase (MacKay, 2003). This is the so-called degeneracy problem of particle methods which is closely tied to the aforementioned curse of dimensionality (Farchi and Bocquet, 2018; Snyder et al., 2008). One way to partly circumvent degeneracy in a sequential setting is to employ resampling techniques (see Li et al., 2015) where a new set of equally weighted particles is drawn based on the weights (i.e., probability masses) of the existing particles. Effectively, resampling tends to reproduce particles with a higher weight while removing particles with lower weight. Resampling thus provides obvious links between particle methods and more heuristic genetic algorithms since the weights can be interpreted as a kind of "fitness" (Chopin and Papaspiliopoulos, 2020). The standard metric (e.g. Särkkä, 2013) for monitoring symptoms of degeneracy is the effective sample size

$$N_{\mathrm{eff}} = \frac{1}{\cancel{\sum_{i=1}^{N}\left(w^{(i)}\right)^2}\,\underset{\sim}{\sum_{i=1}^{N_e}\left(w^{(i)}\right)^2}}\frac{1}{}, \tag{29}$$

where a healthy ensemble of particles would have $\cancel{N_{\mathrm{eff}} = N}\,\underset{\sim}{N_{\mathrm{eff}} = N_e}$ while a completely degenerated ensemble has $N_{\mathrm{eff}} = 1$. This metric can be used to adaptively determine the need for resampling based on a requirement that the ratio $\cancel{N_{\mathrm{eff}}/N}\,\underset{\sim}{N_{\mathrm{eff}}/N_e}$

stays above some "healthy" threshold. Although resampling ensures that weight is more evenly spread among the particles, the occurrence of several identical particles results in sample impoverishment which can in the worst case also lead to a degenerate representation of the posterior. Particle diversity can nonetheless often be rejuvenated implicitly through stochastic terms (jitter) in the dynamical model or more explicitly with a few iterations of Markov Chain Monte Carlo (MCMC; Gilks and Berzuini, 2001).

Having ~~explained importance~~ outlined importance sampling and resampling, the final step is to tie ~~this~~ these steps together with the sequential aspect of particle methods. In the context of both particle filtering and smoothing, importance ~~resampling~~ sampling can be applied sequentially in time as observations become available to the state space model. For the particle filter, this occurs by first drawing particles from the initial prior $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$, then for $k = 1, \ldots, n$ perform SIR:

1. Propagate the particles forward in time through the dynamical model from time $k-1$ to k where new observations are available.

2. Calculate the auto-normalized weights $w_k^{(i)}$ given the current observations.

3. Resample the particles, possibly only if the ratio ~~N_{eff}/N~~ N_{eff}/N_e is below some threshold.

The recipes for implementing most flavors of particle smoothers tend to be a little bit more involved (Särkkä, 2013). Fortunately, the particle batch smoother (PBS) in MuSA is relatively straightforward since in practice it is equivalent to standard sequential importance sampling (SIS) which is just SIR without the resampling step. As such, SIS is quite prone to degeneracy. Nonetheless, an advantage with SIS is that, due to the absence of resampling, the dynamical state history of each particle will be completely consistent with the dynamical model. For the PBS we are only interested in the final (i.e. at time t_n) weights $w_n^{(i)}$ which, when attached to the full state histories, give us an approximation to the full posterior $p(\mathbf{x}_{0:n}|\mathbf{y}_{1:n})$ rather than the filtering distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$. This is advantageous in snow data assimilation since it allows information from observations during the ablation season to propagate backwards in time and influence states in the preceding accumulation season. The final weights can either be computed sequentially using SIS or in a batch update, since these are equivalent when the dynamics are Markovian and the observations are conditionally independent in time. The batch approach used in the PBS is appealing since it can be wrapped around the model allowing the dynamics to evolve freely for the whole data assimilation window from t_0 to t_n without the need for I/O interruptions in the time integration, typically resulting in marked run time acceleration compared to sequential approaches.

~~Having discussed~~ Nwo that we have introduced the particle methods~~in some detail~~, what remains is to ~~outline~~ describe their implementation in MuSA and in particular the choice of proposal distribution. For simplicity, we adopt the standard and simplest approach for particle methods which is to use the prior as the proposal (van Leeuwen and Evensen, 1996). Note that this analogous to what was done in the seminal bootstrap filter of Gordon et al. (1993), and is to our knowledge the only approach considered so far for snow data assimilation. Thereby, recalling that $f(\mathbf{x}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$ and inserting for $q(\mathbf{x}) = p(\mathbf{x})$

then $f(\mathbf{x})/q(\mathbf{x}) = p(\mathbf{y}|\mathbf{x})$ which gives us the following simple expression for the auto-normalized weights

$$w^{(i)} = \frac{f(\mathbf{x}^{(i)})/q(\mathbf{x}^{(i)})}{\sum_{j=1}^{N} f(\mathbf{x}^{(j)})/q(\mathbf{x}^{(j)})} \frac{f(\mathbf{x}^{(i)})/q(\mathbf{x}^{(i)})}{\sum_{j=1}^{N_e} f(\mathbf{x}^{(j)})/q(\mathbf{x}^{(j)})} = \frac{p(\mathbf{y}|\mathbf{x}^{(i)})}{\sum_{j=1}^{N} p(\mathbf{y}|\mathbf{x}^{(j)})} \frac{p(\mathbf{y}|\mathbf{x}^{(i)})}{\sum_{j=1}^{N_e} p(\mathbf{y}|\mathbf{x}^{(j)})}, \tag{30}$$

which is simply the normalized likelihood of the prior particles $\mathbf{x}^{(i)} \sim p(\mathbf{x})$. With the usual Gaussian likelihood employed in MuSA these weights are given by

$$w^{(i)} = \frac{\exp\left(-\frac{1}{2}\left[\mathbf{y}-\widehat{\mathbf{y}}^{(i)}\right]^{\mathrm{T}}\mathbf{R}^{-1}\left[\mathbf{y}-\widehat{\mathbf{y}}^{(i)}\right]\right)}{\sum_{j=1}^{N}\exp\left(-\frac{1}{2}\left[\mathbf{y}-\widehat{\mathbf{y}}^{(j)}\right]^{\mathrm{T}}\mathbf{R}^{-1}\left[\mathbf{y}-\widehat{\mathbf{y}}^{(j)}\right]\right)} \frac{\exp\left(-\frac{1}{2}\left[\mathbf{y}-\widehat{\mathbf{y}}^{(i)}\right]^{\mathrm{T}}\mathbf{R}^{-1}\left[\mathbf{y}-\widehat{\mathbf{y}}^{(i)}\right]\right)}{\sum_{j=1}^{N_e}\exp\left(-\frac{1}{2}\left[\mathbf{y}-\widehat{\mathbf{y}}^{(j)}\right]^{\mathrm{T}}\mathbf{R}^{-1}\left[\mathbf{y}-\widehat{\mathbf{y}}^{(j)}\right]\right)}, \tag{31}$$

where $\widehat{\mathbf{y}}^{(i)} = \mathcal{H}(\mathbf{x}^{(i)})$ are the predicted observations for the i-th particle. In practice, to ensure numerical stability, we first compute the natural logarithm of the weights by using the log-sum-exp trick to avoid potential overflow (Murphy, 2022) and minimize the effects of underflow (Murphy, 2022; Chopin and Papaspiliopoulos, 2020). Subsequently, the weights can be diagnosed by taking the exponential of these stable logarithms.

In the current version of MuSA we apply resampling at each observation timestep for the PF independently of what the effective sample size is. Furthermore the state $\mathbf{x}$ is split into parameters $\mathbf{u}$ and internal states $\mathbf{v}$. The PF in MuSA is then implemented by first drawing the initial parameters as described in the pseudocode in Algorithm 1. For clarity, this pseudo-code omits details such as the log-sum-exp trick (Murphy, 2022) and the way that resampling is performed given that there are several options available (Li et al., 2015).

Algorithm 1 Particle filter

Require: N_e ensemble members, n time steps

1: Draw N_e prior particles from the prior $\mathbf{u}^{(i)} \sim p(\mathbf{u})$ then for $k = 1, \ldots, n$ timesteps: $\mathbf{u}_0^{(i)} \sim p(\mathbf{u}_0)$

2: **for** $k = 1$ **to** n **do**

3:     For all particles ($i = 1, \ldots, n$ $i = 1, \ldots, N_e$) jitter the parameters $\mathbf{u}_k^{(i)} = \mathbf{u}_{k-1}^{(i)} + \boldsymbol{\eta}_{k-1}^{(i)}$ where $\boldsymbol{\eta}_{k-1}^{(i)} \sim \mathrm{N}(\mathbf{0}, \mathbf{Q})$ and run the dynamical model forward in time $\mathbf{v}_k = \mathcal{M}(\mathbf{u}_k, \mathbf{v}_{k-1})$. If this is a timestep with observations $\mathbf{v}_k^{(i)} = \mathcal{M}(\mathbf{u}_k^{(i)}, \mathbf{v}_{k-1}^{(i)})$.

4:     If observations $\mathbf{y}_k$ exist, update the ensemble of particles by calculating the auto-normalized weights $w_k^{(i)} \propto \exp\left(-\frac{1}{2}[\mathbf{y}_k - \widehat{\mathbf{y}}_k^{(i)}]^{\mathrm{T}}\mathbf{R}_k^{-1}[\mathbf{y}_k - \widehat{\mathbf{y}}_k^{(i)}]\right)$ for each particle using the current observations $\mathbf{y}_k$ and predicted observations $\widehat{\mathbf{y}}_k^{(i)} = \mathcal{H}\left(\mathbf{u}_k^{(i)}, \mathbf{v}_k^{(i)}\right)$. If this is a timestep with observations

5:     If observations $\mathbf{y}_k$ exist, resample the particles based on their weights $w_k^{(i)}$ and then reset all the weights to be equal, i.e. $w_k^{(i)} = 1/N$.

6: **end for**

Note that the particles will all have a posterior weight equal to $1/N_e$ due to the resampling which happens at each observation timestep. Combined with the transition density, these weights form the prior for the next observation time. This explains why no weights from the prior enter into the posterior weight calculation, since these are all equal and so cancel in the auto-normalization. For the PF, the DAW can be understood to be the time interval between two neighboring observation times. The

prior mean and variance of the internal states can be obtained by taking the (unweighted) ensemble means and variances of the particle trajectories $\mathbf{v}_k^{(i)}$ in the DAW before the resampling step. The posterior means and variances of the internal states can be obtained by taking the (unweighted) ensemble mean and variance of the particle trajectories $\mathbf{v}_k^{(i)}$ in the DAW after the resampling step.

In MuSA, the PBS is even more straightforwards to implement than the PF. Pseudocode for the PBS algorithm as implemented in MuSA is outlined in Algorithm 2. Here too we have emphasized clarity in the pseudocode and omit details from the actual code such as the log-sum-exp trick .

Algorithm 2 Particle batch smoother

Require: N_e ensemble members, n time steps

  1: Draw N_e prior particles from the prior $\mathbf{u}^{(i)} \sim p(\mathbf{u})$

  2: **for** $k = 1$ **to** n **do**

  3:     For all particles ($i = 1, \ldots, N_e$) run the dynamical model forward in time, $\mathbf{v}_k^{(i)} = \mathcal{M}(\mathbf{u}^{(i)}, \mathbf{v}_{k-1}^{(i)})$.

  4:     If observations $\mathbf{y}_k$ exist, append these and the predicted observations $\widehat{\mathbf{y}}_k^{(i)} = \mathcal{H}(\mathbf{u}^{(i)}, \mathbf{v}_k^{(i)})$ for each particle to the batch of observations $\mathbf{y}$ and predicted observations $\widehat{\mathbf{y}}^{(i)}$, respectively.

  5:     If $k = n$, calculate the weights for the entire DAW using the batch of observations and predicted observations $w^{(i)} \propto \exp\left(-\frac{1}{2}[\mathbf{y} - \widehat{\mathbf{y}}^{(i)}]^{\mathrm{T}} \mathbf{R}^{-1}[\mathbf{y} - \widehat{\mathbf{y}}^{(i)}]\right)$.

  6: **end for**

3.5 Ensemble Kalman methods

The ensemble Kalman filter (EnKF) was originally proposed by Evensen (1994) as a Monte Carlo version of the original Kalman filter (KF; see Jazwinski, 1970; Särkkä, 2013). The original KF assumes that all distributions involved in filtering are Gaussian and that both the dynamical and observation models are linear. These assumptions come on top of the usual filtering assumptions of Markovian state dynamics and conditionally independent observations. If all these assumptions are met, the exact filtering distribution turns out to be a Gaussian of the form $p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{x}_k|\mathbf{m}_k, \mathbf{P}_k)$ and there is a set of closed form (i.e. analytical) equations for the mean $\mathbf{m}_k$ and covariance $\mathbf{P}_k$, which completely defines this Gaussian filtering distribution, known as the KF equations (Särkkä, 2013). Geoscientific models are almost never linear so these equations can usually not be applied directly in practice. Even when they can, the need to explicitly store and update the state covariance matrix $\mathbf{P}_k$ with dimensions $m \times m$ can be computationally prohibitive in high dimensional problems typical in geoscience (Carrassi et al., 2018).

There exist several modified versions of the KF such as the extended KF (EKF) based on Taylor series approximations and the unscented KF (UKF) based on the unscented transform that can both partly circumvent the linearity assumption (Särkkä, 2013). Since both the EKF and UKF also require an explicit covariance matrix update and are considerably more challenging to implement than the KF, they have only rarely been used for geoscientific DA. Instead, since its introduction by Evensen (1994), it is the EnKF that has become one of the methods of choice for DA. The reasons for this are that the

EnKF is relatively straightforward to implement and understand (Katzfuss et al., 2016), it can handle non-linear models, the state covariance evolves implicitly with the ensemble, and it has proven to work well in very high dimensional operational applications both in meteorology and oceanography (see Carrassi et al., 2018, for a review). Although it makes the same Gaussian linear assumptions as the KF, in practice it can often still function well when these assumptions are strongly violated thanks to techniques such as Gaussian anamorphosis (Bertino et al., 2003) and iterations (Emerick and Reynolds, 2013). The ensemble Kalman approach can also be applied to solve smoothing problems as first shown by van Leeuwen and Evensen (1996). These smoothers, particularly the so-called ensemble smoother (ES), play an important role in solving reanalysis problems that arise in both land surface and snow data assimilation (Dunne and Entekhabi, 2005; Durand and Margulis, 2006).

The EnKF was one of the first schemes to be used for snow data assimilation. Slater and Clark (2006) used the EnKF to assimilate SWE data into a conceptual snow model at several sites in Colorado, leading to marked improvements in SWE estimates compared to both the control model runs and interpolated observations. Clark et al. (2006) used a snow depletion curve to assimilate synthetic satellite retrievals of FSCA into a conceptual snow hydrology model of a catchment in Colorado, leading to minor improvements in simulated streamflow. Durand and Margulis (2006) jointly assimilated synthetic satellite retrievals of brightness temperature and albedo into the SSiB3 model using the EnKF at a site in the California Sierra Nevada to yield marked improvements compared to the open loop. Andreadis and Lettenmaier (2006) used the EnKF to assimilate MODIS FSCA into the VIC model for the Snake River basin in the Pacific Northwest region of the USA, improving model estimates of both FSCA and SWE.

Following on from these initial studies there have been several more recent snow DA applications using the EnKF. In a synthetic experiment in northern Colorado, De Lannoy et al. (2010) showed how the EnKF can be used to assimilate coarse scale passive microwave SWE observations into high resolution runs of the Noah land surface model, leading to large reductions of error in SWE estimation compared to the open loop. In a follow up study~~at the same site~~, De Lannoy et al. (2012) jointly assimilated real MODIS FSCA and AMSR-E SWE retrievals into the Noah model to demonstrate the complementary nature of these types of satellite observations for SWE estimation. Magnusson et al. (2014) used the EnKF to assimilate data from ground-based stations in the Swiss Alps and showed that assimilating fluxes (snowmelt and snowfall), rather than directly assimilating SWE data, improved the SWE estimation. The study of Huang et al. (2017) demonstrated that the assimilation of in-situ SWE data into a hydrological model with the EnKF could improve streamflow simulations for several basins in the western USA. Stigter et al. (2017) performed a joint assimilation of in-situ snow depth data and FSCA satellite retrievals with the EnKF to optimize parameters and subsequently estimate the climate sensitivity of SWE and snowmelt runoff for a Himalayan catchment. More recently, Hou et al. (2021) showed how machine learning techniques can be used to construct empirical observation operators to assimilate satellite-based FSCA with the EnKF.

The ensemble smoother (ES), the batch smoother version of the EnKF, has also been used extensively for snow DA, mainly for reanalysis. Durand et al. (2008) were the first to demonstrate the value of the ES for snow reanalysis by noting that it could be used ~~in a more robust Bayesian~~ as a new approach to solve the traditional problem of snow reconstruction. They showed substantial improvements in SWE estimation for the posterior compared to the prior after having assimilated synthetic FSCA observations into SSiB3. A key advantage of the ES over the EnKF is that it allows information to propagate backwards in

time, so that observations made in the ablation season can update the preceding accumulation season. Following up this study with a real experiment assimilating Landsat FSCA retrievals into SSiB3 using the ES at a basin in the California Sierra Nevada,

680 Girotto et al. (2014a) demonstrated explicitly how snow reanalysis using the ES is a more robust probabilistic generalization of the traditional deterministic approach to snow reconstruction. Subsequently, Girotto et al. (2014b) applied the same ES snow reanalysis framework to produce a multi-decadal high resolution snow reanalysis for the Kern River watershed in the California Sierra Nevada. Oaida et al. (2019) used the ES to assimilate MODIS FSCA observations into the VIC model to produce a kilometer-scale snow reanalysis over the entire western USA. Aalstad et al. (2018) introduced an iterative version

685 of the ES, the ensemble smoother with multiple data assimilation scheme (ES-MDA; Emerick and Reynolds, 2013), for snow reanalysis and showed how this could outperform both the ES and the PBS when assimilating FSCA retrievals at sites on the high-Arctic Svalbard archipelago. In this iterative scheme the prior ensemble moves gradually to the posterior ensemble through a tempering procedure (see Stordal and Elsheikh, 2015, and references therein). The iterations thus mitigate the impact of the linearity assumption inherent in ensemble Kalman methods (Evensen, 2018), typically leading to marked improvements

690 compared to the ES for nonlinear models without the risk of degeneracy associated with particle methods as the curse of dimensionality rears its head (Snyder et al., 2008; Pirk et al., 2022).

A derivation of the KF equations, which the EnKF and ES are largely based on, is beyond the scope of this work. Full multivariate Bayesian derivations of these equations can be found on page 197 of Jazwinski (1970) and Section 4.3 of Särkkä (2013). The EnKF itself is thoroughly derived in Evensen (2009) and Evensen et al. (2022). There are actually several variants

695 of the ensemble Kalman analysis step (see Carrassi et al., 2018). In MuSA we use the so-called stochastic rather than deterministic (or square-root) implementation. This stochastic formulation ~~was proposed by Burgers et al. (1998) who suggested adding stochastic~~ adds perturbations to the predicted observations to ensure adequate ensemble spread ~~, but it was recently corrected by van Leeuwen (2020) who showed that perturbations should actually be applied to the predicted observations. In practice, due to the symmetry of the perturbations, this recent correction should have minimal impact but it provides much needed~~

700 ~~clarification and is consistent~~ and consistency with the underlying Bayesian theory (Burgers et al., 1998; van Leeuwen, 2020).

Here we present the equations for both the stochastic EnKF (van Leeuwen, 2020) and the ES (van Leeuwen and Evensen, 1996), as well as their iterative versions based on the multiple data assimilation (MDA) scheme of Emerick and Reynolds (2012, 2013), in the form that they are used in MuSA. The iterative versions tend to perform better than their non-iterative counterparts for non-linear problems (Evensen, 2018). Let N_a denote the number of assimilation cycles (iterations) performed in a pseudo (rather than model) time. For the standard EnKF and ES we set $N_a = 1$ while for their iterative variants $N_a > 1$, typ-

705 ically with $N_a = 4$ (Emerick and Reynolds, 2013; Aalstad et al., 2018). The superscript ℓ is used to index these iterations. Let $\mathbf{U}^{(\ell)} = \left[\mathbf{u}^{(1)(\ell)}, \ldots, \mathbf{u}^{(i)(\ell)}, \ldots, \mathbf{u}^{(N)(\ell)}\right]$ denote the $m_p \times N$ $\mathbf{U}^{(\ell)} = \left[\mathbf{u}^{(1)(\ell)}, \ldots, \mathbf{u}^{(i)(\ell)}, \ldots, \mathbf{u}^{(N_e)(\ell)}\right]$ denote the $m_p \times N_e$ parameter matrix containing the ensemble ($i = 1, \ldots, n$) of parameter vectors $\mathbf{u}^{(i)(\ell)}$ for iteration ℓ. Recall that the subset of these parameters that are physically bounded have undergone the relevant analytic transformations for Gaussian anamorphosis, and

710 the corresponding inverse transforms are applied back to physical space when these are passed through the dynamical model (see Aalstad et al., 2018). Similarly, let $\widehat{\mathbf{Y}}^{(\ell)} = \left[\widehat{\mathbf{y}}^{(1)(\ell)}, \ldots, \widehat{\mathbf{y}}^{(i)(\ell)}, \ldots, \widehat{\mathbf{y}}^{(N)(\ell)}\right]$ $\widehat{\mathbf{Y}}^{(\ell)} = \left[\widehat{\mathbf{y}}^{(1)(\ell)}, \ldots, \widehat{\mathbf{y}}^{(i)(\ell)}, \ldots, \widehat{\mathbf{y}}^{(N_e)(\ell)}\right]$ denote the predicted observation matrix containing the ensemble of predicted observations $\widehat{\mathbf{y}}^{(i)(\ell)} = \mathcal{H}\left(\mathbf{u}^{(i)(\ell)}, \mathbf{v}^{(i)(\ell)}\right)$. ~~Then~~

Adopting this notation, the stochastic ensemble Kalman methods ~~proceed by first drawing the initial parameters~~ in MuSA are implemented as described in the pseudocode in algorithm 3. For clarity, this pseudo-code also omits details such as how the matrix inversion for the ensemble Kalman gain in Equation (32) is carried out. For this we follow the pseudo-inverse and rescaling approach outlined in Appendix A of Emerick and Reynolds (2012).

Algorithm 3 Ensemble Kalman analysis with multiple data assimilation

---

Require: N_e ensemble members, N_a assimilation cycles

1: Draw $i = 1, \ldots, N_e$ prior particles from the prior $\mathbf{u}^{(\ell=0)} \sim p(\mathbf{u})$, then for $\ell = 0 : N_a$ iterations: $\mathbf{u}^{(i)(\ell=0)} \sim p(\mathbf{u})$

2: **for** $\ell = 0$ **to** N_a **do**

3:    Run the dynamical model forward in time for all ensemble members i to obtain the internal states $\mathbf{v}^{(i)(\ell)}$ and predicted observations $\widehat{\mathbf{y}}^{(i)(l)}$ corresponding to the ensemble of parameter vectors $\mathbf{u}^{(i)(\ell)}$. If $\ell < N_a$ perform the ensemble Kalman analysis step

$$\mathbf{U}^{(\ell+1)} = \mathbf{U}^{(\ell)} + \mathbf{K}^{(\ell)}\left[\mathbf{Y} - \left(\widehat{\mathbf{Y}}^{(\ell)} + \boldsymbol{\mathcal{E}}_\alpha^{(\ell)}\right)\right],$$

4:    **if** $\ell < N_a$ **then**

5:       Generate a $d \times N_e$ matrix of inflated observation errors $\boldsymbol{\mathcal{E}}_\alpha^{(\ell)} = \sqrt{\alpha^{(\ell)}}\mathbf{R}^{1/2}\boldsymbol{\epsilon}^{(\ell)}$ where $\mathbf{Y}$ is an $d \times N$ matrix containing N copies of the observation vector $\mathbf{y}$ while the observation error term is given by

$$\boldsymbol{\mathcal{E}}_\alpha^{(\ell)} = \sqrt{\alpha^{(\ell)}}\mathbf{R}^{1/2}\boldsymbol{\epsilon}^{(\ell)},$$

in which $\boldsymbol{\epsilon}^{(\ell)}$ is an $d \times N$ matrix containing draws from a standard Gaussian N(0,1) $\boldsymbol{\epsilon}^{(\ell)} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$ is an $d \times N_e$ matrix and $\alpha^{(\ell)} = N_a$ is the observation error inflation coefficient. The (ensemble) Kalman gain $\mathbf{K}^{(\ell)}$ is an $m_p \times d$ matrix given by

$$\mathbf{K}^{(\ell)} = \mathbf{C}_{\mathbf{U}\widehat{\mathbf{Y}}}^{(\ell)}\left(\mathbf{C}_{\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}}^{(\ell)} + \alpha^{(\ell)}\mathbf{R}\right)^{-1},$$

where, in these ensemble Kalman methods, the model-based covariance matrices are estimated from the ensemble such that

6:       Compute the $m_p \times d$ ensemble parameter-predicted observation covariance matrix is given by

$$\mathbf{C}_{\mathbf{U}\widehat{\mathbf{Y}}}^{(\ell)} = \frac{1}{N}\mathbf{U}^{(\ell)'}\widehat{\mathbf{Y}}^{(\ell)'\mathrm{T}},$$

while the $d \times d$ predicted observation covariance matrix is given by

$$\mathbf{C}_{\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}}^{(\ell)} = \frac{1}{N}\widehat{\mathbf{Y}}^{(\ell)'}\widehat{\mathbf{Y}}^{(\ell)'\mathrm{T}},$$

$\mathbf{C}_{\mathbf{U}\widehat{\mathbf{Y}}}^{(\ell)} = \frac{1}{N}\mathbf{U}^{(\ell)'}\widehat{\mathbf{Y}}^{(\ell)'\mathrm{T}}$ where primes $(\cdot)'$ denote deviations from the ensemble mean.

7:       Compute the $d \times d$ ensemble predicted observation covariance matrix $\mathbf{C}_{\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}}^{(\ell)} = \frac{1}{N}\widehat{\mathbf{Y}}^{(\ell)'}\widehat{\mathbf{Y}}^{(\ell)'\mathrm{T}}$

8:       Compute the $m_p \times d$ ensemble Kalman gain matrix

$$\mathbf{K}^{(\ell)} = \mathbf{C}_{\mathbf{U}\widehat{\mathbf{Y}}}^{(\ell)}\left(\mathbf{C}_{\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}}^{(\ell)} + \alpha^{(\ell)}\mathbf{R}\right)^{-1}. \tag{32}$$

9:       Perform the ensemble Kalman analysis step to update the parameter ensemble $\mathbf{U}$

$$\mathbf{U}^{(\ell+1)} = \mathbf{U}^{(\ell)} + \mathbf{K}^{(\ell)}\left[\mathbf{Y} - \left(\widehat{\mathbf{Y}}^{(\ell)} + \boldsymbol{\mathcal{E}}_\alpha^{(\ell)}\right)\right], \qquad \textbf{26} \tag{33}$$

where $\mathbf{Y}$ is an $d \times N_e$ matrix containing N_e copies of the observation vector $\mathbf{y}$.

Recall that for $N_a = 1$ we recover the non-iterative stochastic EnKF and ES, while for $N_a > 1$ we are using iterative versions of these schemes that involve multiple data assimilation with inflated observation errors. ~~This~~ The term "multiple data assimilation" refers to the assimilation of the same data multiple times rather than an assimilation of different types of data (joint assimilation). We speak of inflated observation errors since the role of the coefficients $\alpha^{(\ell)}$ is to inflate the observation error covariance $\mathbf{R}$ in the Kalman gain $\mathbf{K}^\ell$ as well as the observation error term $\boldsymbol{\mathcal{E}}_\alpha^{(\ell)}$. This inflation is tantamount to tempering the likelihood as discussed by ~~Stordal and Elsheikh (2015); van Leeuwen et al. (2019)~~ Stordal and Elsheikh (2015) and van Leeuwen et al. (2019) which explains why these iterative schemes perform better than their non-iterative counterparts for non-linear models in that they involve a more gradual transition from the prior to the posterior. Despite what the name might suggest, this multiple data assimilation approach does not actually violate the consistency of Bayesian inference by using the data more than once due to the way the observation error inflation is constructed, particularly due to the constraint that ~~$\sum_{\ell=1}^{N_a}] = 1$~~ $\sum_{\ell=1}^{N_a} 1/\alpha^{(\ell)} = 1$. Simply stated, this constraint ensures consistent results with a linear model since by construction we get the same result by assimilating the data once with the original uninflated ($\alpha = 1$) observation errors as assimilating the same data multiple times with an inflated ($\alpha > 1$) observation errors. With a nonlinear model, practice has shown that these iterations of the ensemble Kalman analysis ensure that the approximate posterior is closer to the true posterior than if a more conventional single uninflated iteration is used (Evensen, 2018). It is possible to satisfy ~~this constraint~~ the constraint on the $\alpha^{(\ell)}$ both with uniform and non-uniform inflation coefficients (c.f. Evensen, 2019). For simplicity, following Emerick and Reynolds (2013), we currently opt for former as a default in MuSA by setting $\alpha^{(\ell)} = N_a$ ~~set $\alpha^{(\ell)} = N_a$~~ for all ℓ by default while allowing for ~~the latter~~ non-uniform coefficients as an option.

To keep the notation as simple as possible, we have not explicitly introduced time indices for these ensemble Kalman methods. Instead we can simply note that for the iterative and non-iterative EnKF the for loop above runs inside another outer loop that runs across all observation times. That is to say, the ℓ loop is run several times sequentially for multiple DAWs where each window is from one observation time to the next, such that the posterior parameters and internal states obtained at $\ell = N_a$ for the current DAW that ran up to the current observation time becomes the initial prior at $\ell = 0$ for the next DAW that runs from the current observation time up to the next observation time. Recall that the parameters also undergo jitter dynamics when filtering.

For the non-iterative and iterative ensemble batch Kalman smoothers used in MuSA, namely the ES and the ES-MDA, there is no longer an outer time loop encapsulating the iterations. Instead, the DAW is the entire water year. In these smoother schemes, the parameters do not undergo any jitter. As such, the entire prior and posterior ensemble of state trajectories for a given water year will be consistent with the prior and posterior ensemble of ~~parameters, respectively, that are both static for the water year in question~~ static parameters for the entire water year. Recall that each water year is ~~treated~~ typically updated independently in MuSA which currently focuses on seasonal snow.

4   Data and experimental setup

In the following section, we illustrate the capabilities of MuSA using a case study and perform a benchmarking of the imple-
mented data assimilation algorithms. We developed two data assimilation experiments in the 55 ha Izas experimental catchment
(Revuelto et al., 2017) in the Spanish Pyrenees (see Figure 2). The first experiment shows the capabilities of MuSA to assim-
ilate hyper-resolution products in both a single cell and distributed fashion, whereas the second demonstrates the capabilities
to develop joint assimilation experiments. ~~First, we~~ Note that all the variables that we assimilate in these experiments are state
variables in FSM2. We generated hourly meteorological forcing at 5 m resolution over the Izas catchment ~~by means of~~ using
the MicroMet meteorological distribution system (Liston and Elder, 2006b) ~~, forced by~~ to downscale the ERA5 atmospheric
reanalysis (Hersbach et al., 2020).
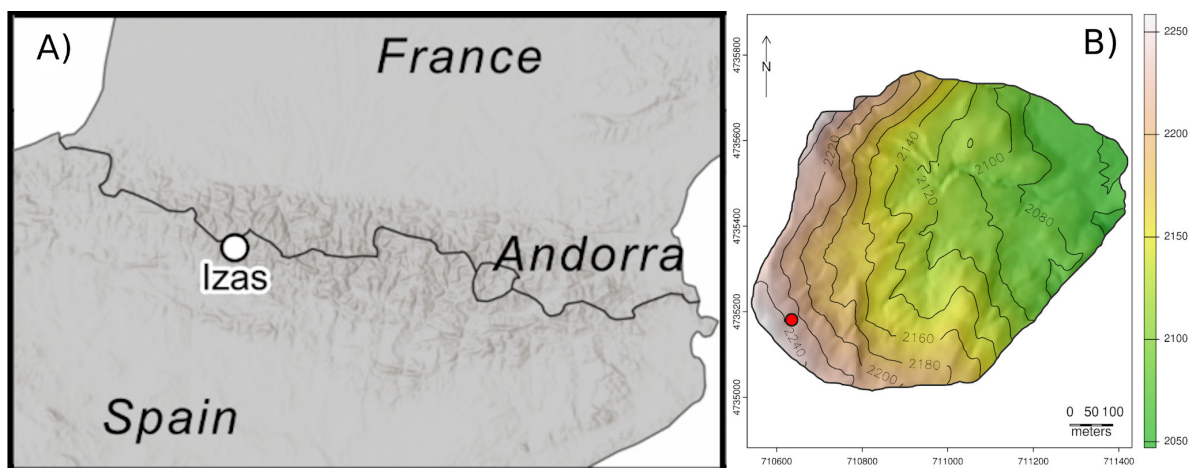
Figure 2. Location of the Izas experimental basin in the Pyrenees (A) and topography (B). The red circle indicates the location of the cell
used for the intercomparison of algorithms described below.

4.1   ~~Drone data~~ Single cell and distributed assimilation ~~experiment~~ of drone-based snow depth retrievals

For the first experiment, we ran MuSA to assimilate snow depth retrievals obtained from a fixed wing drone (Revuelto et al.,
2021b, a), over two different snow seasons (2018/2019 and 2019/2020) in the Izas catchment using different data assimilation
algorithms. The drone surveys were not equally distributed in time along the seasons, and the surveys present spatial gaps
in some of the cells. The whole drone dataset is composed of 18 temporally distributed snow depth maps retrieved at 5m
spatial resolution. Due to the high-resolution of the observations, we expect MuSA to be able to implicitly reproduce the wind
redistribution patterns by ~~inflating the uncertainty of the forcing~~ modifying the precipitation and temperature at grid cell scale
to compensate for the deposition and removal of wind-blown snow.

The simulation ensemble was composed of ~~200~~ $N_e = 200$ particles. The generation of the prior ensemble for both the single cell and distributed simulation was developed by perturbing only the temperature and precipitation fields. The temperature was perturbed with a constant additive bias parameter, randomly drawn from a normal distribution defined by its mean ($\mu = 0$) and standard deviation ($\sigma = 2$). Similarly, the precipitation was perturbed by a multiplicative fixed in time bias parameter, drawn from a log-normal distribution defined by the mean ($\mu = 0$) and standard deviation ($\sigma = 0.63$) of the underlying normal distribution. These prior distributions were defined based on prior predictive testing (not shown here). As such, these are weakly informative priors (c.f. Banner et al., 2020) in that they produce predictions that we expect a-priori (i.e., without looking at the data) to be in the right correct of magnitude while obeying physical bounds. The variances of the observation errors were set to 0.04 [m^2] and was estimated from comparison with terrestrial laser scanning snow depth maps (Revuelto et al., 2021b, a). In the case of the single cell comparison, we followed two different resampling strategies, the bootstrap and redraw from a normal approximation of the posterior (see Section 2). In the iterative versions of the ensemble Kalman based approaches we fixed the number of assimilation cycles to $N_a = 4$ as a compromise between computational cost and performance. The former is directly proportional to N_a while the latter converges to an optimum as N_a increases. This choice of N_a is also in line with sensitivity analyses performed elsewhere (Emerick and Reynolds, 2013; Aalstad et al., 2018; Evensen, 2018).

The single cell simulations were performed in a cell located in a topographic concavity within the catchment, and therefore with exceptional accumulations and snow depth up to 6 m due to wind redistribution and preferential deposition (Comola et al., 2019). This is a challenge for the data assimilation process, as the observations fell very far from the central prior range of the ensemble of snow depths predicted by FSM2. The distributed simulation was developed in a supercomputing cluster using 20 nodes with 10 cores each. For this case study, we choose the most computationally efficient data assimilation algorithm, namely the PBS, to speed up the calculations. We subsequently compared the spatial distribution of the snowpack with one of the snow depth maps (03/02/2020) derived from the drone surveys that were not included in the assimilation.

4.2 Joint ~~satellite data~~ assimilation ~~experiment~~ of satellite-based LST and FSCA retrievals

For the second experiment, we performed a coarser 1 km resolution experiment over the same area. We developed a single cell joint assimilation experiment, updating the FSM2 simulations with FSCA and land surface temperature (LST) retrievals from the MODIS sensor on board the Terra satellite, by using version 6 of the products MOD11A1 ~~(Hall and Riggs, 2016)~~ (Wan et al., 2015) and MOD10A1 ~~(Wan et al., 2015)~~ (Hall and Riggs, 2016), respectively. The selected pixel for the MODIS retrievals was the one whose centroid fell closest to the Izas catchment centroid. While the FSCA retrievals from MODIS have a 500 m spatial resolution, the LST products have a resolution of 1000 m. To assimilate this information into FSM2 using the MuSA platform, we aggregated the FSCA products to match the 1 km LST grid. The forcing was generated by aggregating the 5 m forcing from the previous distributed data assimilation experiment. The variances of the observation errors were set to 0.15 and 10 [K^2] for the FSCA and the LST, respectively. The data assimilation experiment was done using the ES-MDA scheme. This data assimilation scheme was chosen because it ~~showed a better accuracy compared with~~ performed better than the other algorithms (not shown here), that in some cases were not able to meaningfully update the simulations with the observations at all.

The number of iterations was again set to $N_a = 4$. The ensemble was composed of ~~$N = 300$~~ $N_e = 300$ particles by perturbing all the forcing variables (temperature, precipitation, short and longwave radiation, wind speed, relative humidity and surface air pressure) with normal additive or lognormal multiplicative random noise depending on whether or not the forcing variable has negative support so as to not generate physically impossible values.

4.3 Computational benchmarks

Finally, we developed a single cell benchmark, to measure the computational cost in terms of wall clock time for each algorithm. The benchmark was developed in a local machine with an Intel(R) Core(TM) i5-1145G7 processor and 32GB of memory running an Ubuntu 20.04 system. The comparison was performed using 100, 200 and 300 particles and four iterations for the iterative ensemble Kalman approaches in a single cell, assimilating drone snow depth retrievals at a random location in the Izas catchment. The reported values of the benchmarks are the average of 10 MuSA runs, and includes the FSM2 compilation time ($\simeq$2 seconds using the GNU Fortran compiler 10.3.0 in the aforementioned local machine) which is negligible compared with the whole run.

5 Results

5.1 Single cell and distributed ~~hyper-resolution drone data~~ assimilation of drone-based snow depth retrievals

The results show how the performance of the different data assimilation algorithms differs, even with the same initial conditions and experimental setup~~(Figure 3 and Figure 4, single cell test comparison~~, when the posterior ensembles are compared against the assimilated snow depth observations (Table 1, Figures 3&4). The ~~deterministic open loop~~ reference simulation exhibited a root mean squared error (RMSE) compared with the observations of ~~3.1~~ 3.18 m, higher than the posterior ~~average~~ mean of any of the algorithms implemented in MuSA~~as~~, except for the PF with bootstrap resampling which collapses early in the simulation. The results are shown in Table 1:

	Scheme							
Metric	~~PF-c~~ Ref	~~PF-b~~	PF-r	EnKF	EnKF-MDA	PBS	ES	ES-MDA
RMSE [m]	~~1.51 0.04~~ 3.18	~~0.98~~ 4.46	0.34	~~0.17~~ 1.65	~~0.16~~ 0.73	~~0.09~~ 0.63	0.7	0.39

Table 1. ~~Evaluation metrics~~ RMSE for the reference run (Ref), particle filter with bootstrap resampling (PF-b), particle filter with redraw resampling (PF-r), ensemble Kalman filter (EnKF), ensemble Kalman filter with multiple data assimilation (EnKF-MDA), particle batch smoother (PBS), ensemble smoother (ES), and ensemble smoother with multiple data assimilation (ES-MDA). These errors were computed using the assimilated drone-based snow depth observations as the truth and using the posterior ensemble mean ~~based on~~ as the ~~assimilated observations~~ estimate from the respective DA schemes. All the schemes were run with $N_e = 200$ particles and the MDA schemes used $N_a = 4$ iterations.

The ~~particle filter~~ PF collapsed early in the singe cell simulation where all the weight is shared by just a few and eventually a single particle (see Figure 3), due to the large difference between the observations and the ensemble of predictions at the first assimilation step. The particle filter with ~~redraw~~ redraw-based resampling allowed the assimilation process to recover from the initial collapse through particle rejuvenation, leading to a more realistic non-degenerate ensemble simulation. Conversely, the ensemble Kalman-based approaches were less prone to ensemble degeneracy. Here, the ~~ensemble Kalman filter~~ EnKF produced unsatisfactory results when the observations fall very far from the ~~ensemble. However,~~ posterior ensemble in the first season, improving considerably in the second. Note, however, that we did not observe ~~this~~ the same issue with the ~~iterative version of the ensemble Kalman filter~~ EnKF-MDA.
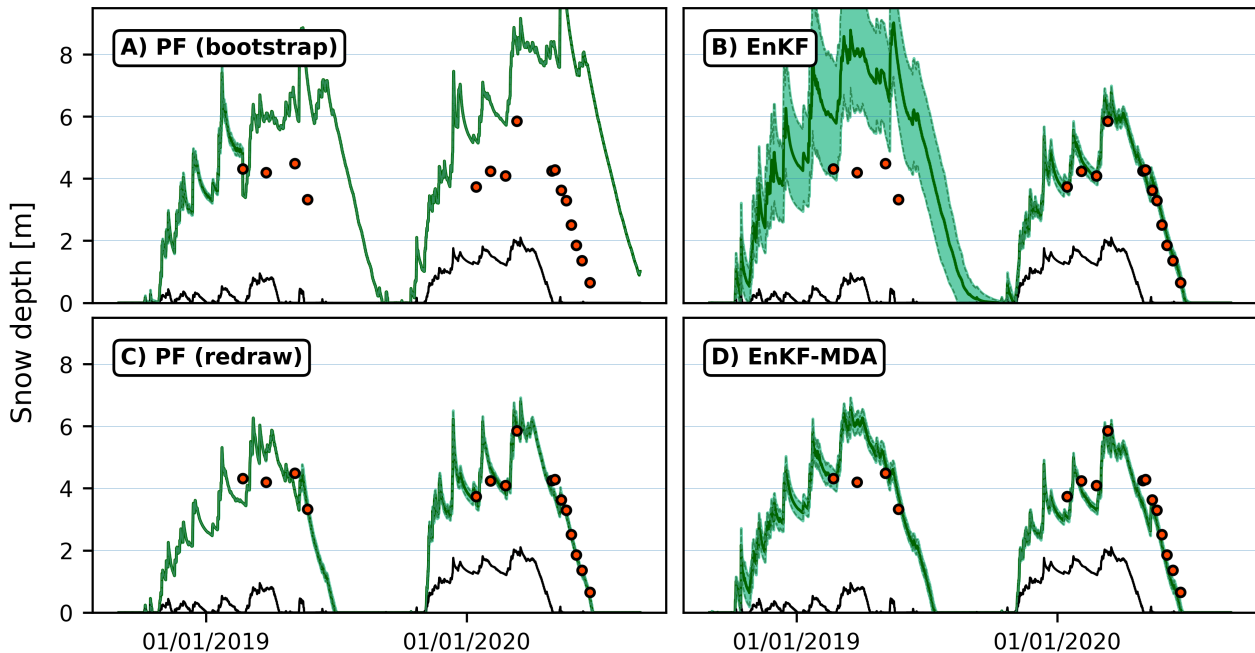
Figure 3. Intercomparison of ~~filter based~~ filter-based data assimilation experiments in the single cell of the Izas catchment highlighted in Figure 2. The black line is a deterministic (no perturbations) open-loop simulation for reference and the red dots are the observations, the ~~light blue lines are the ensemble of posterior state while the~~ dark ~~blue~~ green line ~~is~~ (shading) shows the ~~ensemble~~ posterior mean (± 1 standard deviation).

Figure 4 shows the results of the experiment using the smoother based approaches. In this particular case, the posterior snow depth time series generated by using the PBS are much closer to the observations than when using particle filters with the bootstrap resampling (Figure~~3~~ 3A). In addition, due to the absence of resampling, the PBS did not generate discontinuities in the series that can be observed in the particle filters posteriors (Figure 3). The ensemble (Kalman) smoothers also outper-

formed their filter counterpart in this particular case. Moreover, the ~~iterative ensemble (Kalman) smoother~~ ES-MDA produced simulations closer to the observations compared to the other smoothers as confirmed in Table 1~~)~~.
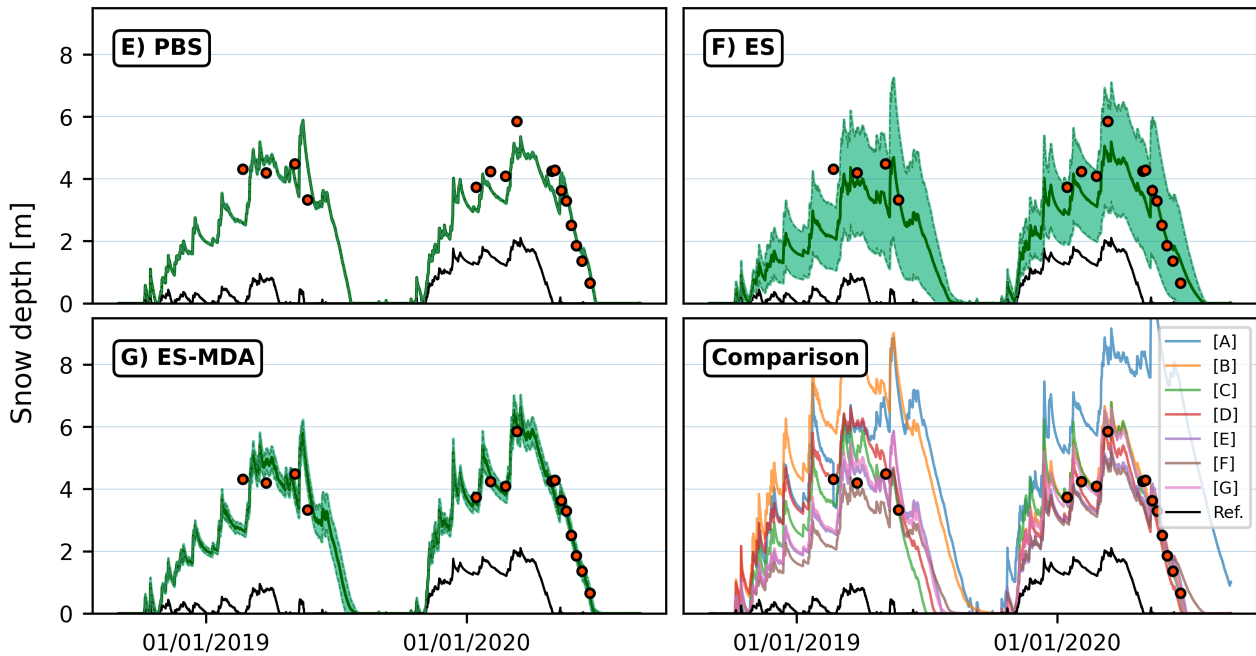
Figure 4. Intercomparison of ~~smoother based~~ smoother-based data assimilation experiments and intercomparison of the posterior mean of all the data assimilation algorithms (letters A-G correspond to the panels in which the individual schemes are shown in Figures 3&4) in the single cell of the Izas catchment highlighted in Figure 2. The black line is a deterministic (no perturbations) open-loop simulation for reference and the red dots are the observations, the ~~light blue lines are the ensemble of posterior state while the~~ dark ~~blue~~ green line ~~is the ensemble mean. PF-c, PF-r, I and OL refers to particle filter (classic resampling~~shading) ~~, particle filter (redraw from~~ shows the posterior ~~resampling~~mean ($\pm$ 1 standard deviation)~~, iterative versions of the algorithms and the open loop, respectively.~~

There is an obvious improvement in the snow depth spatial distribution patterns after running MuSA compared with the ~~open-loop~~ reference simulations for the hyper-resolution distributed simulation as shown in Figure 5. The updated SWE products exhibited very consistent spatial patterns compared with the independent (i.e. non-assimilated) drone-based snow depth map, while the ~~open loop~~ reference simulations exhibited a much lower and less realistic spatial variability. Similarly, the updated FSM2 snow depth simulations were very consistent with the non-assimilated snow depth maps with a coefficient of determination of ~~R²=0.96 while the open loop~~ $R^2 = 0.96$ while the reference simulation exhibited an ~~R²=0.03~~ $R^2 = 0.03$.

Figure 6 shows the spatial distribution of the ~~weighted~~ posterior mean temperature bias ~~(based on a normal prior distribution, i.e. an additive factor) and precipitation bias (based on log-normal prior distribution, i.e. a multiplicative factor)~~ and precipitation multiplier over the Izas catchment after the assimilation of the drone-based snow depth retrievals. The distribution of the pertur-
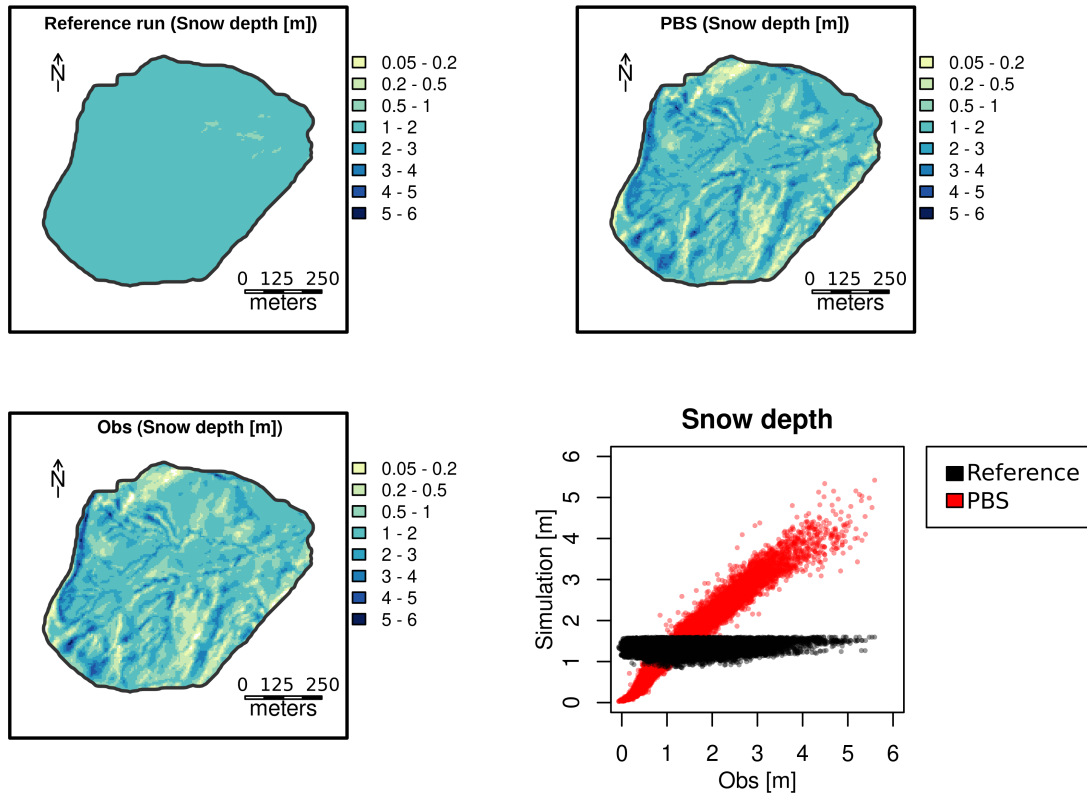
Figure 5. ~~Open loop FSM~~ Snow depth estimates for Izas on the 03/02/2020: reference simulation ~~and updated FSM~~ (top left), PBS posterior mean simulation (~~upper panels~~top right)~~. Independent drone snow depths compared to the open loop~~, independent drone-based retrievals (~~black~~bottom left)~~and~~, evaluation of the ~~posterior~~ simulations (black: reference, red: PBS) ~~,respectively~~ relative to the drone-based retrievals (~~lower panels~~bottom right).

bation parameters showed spatial patterns consistent with the topography of Izas in both of the variables, ranging from -4 to 2 [K] in the case of the additive temperature bias and from 0.5 to 3 in the case of the ~~multiplicative precipitation bias.~~precipitation multiplier.

845 ## 5.2 ~~Coarser-scale joint satellite data~~ Joint assimilation **of satellite-based LST and FSCA retrievals**

The ~~results of the~~result of jointly assimilating LST and FSCA ~~joint assimilation~~ is shown in Figure 7. As expected, the difference between the observations and the updated simulations was reduced, compared with the ~~open loop~~reference simulations. The root mean squared error (RMSE) between the observations and the ~~open loop~~reference LST was 7.2 [K] compared to 5.1 [K] for the updated simulations. Similarly, the RMSE between the observations and the ~~open loop~~ reference FSCA was 0.38, 850 compared to 0.14 for the updated simulations. Such modifications of the FSCA and LST state variables resulted in a completely
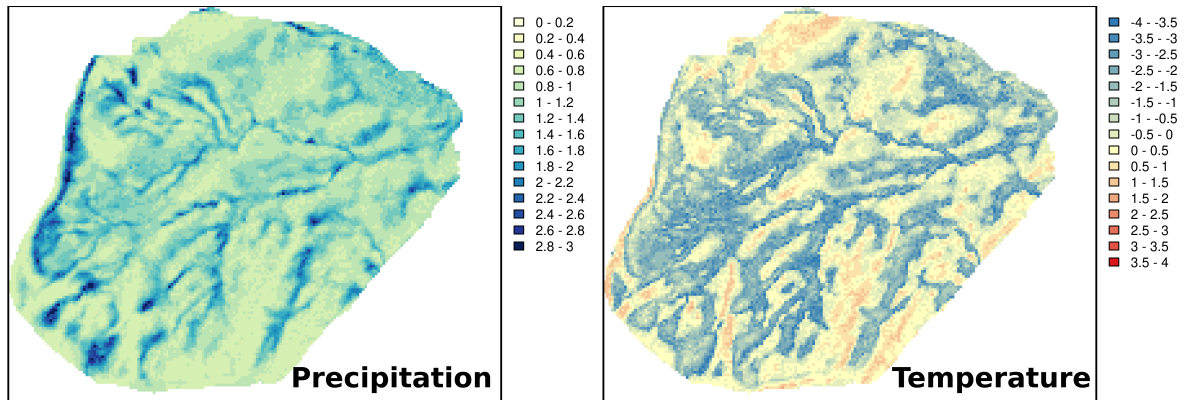
Figure 6. Posterior perturbation parameters for the precipitation [non-dimensional] (multiplicative) and temperature [K] (additive).

different SWE simulation, that reached values up to 900 mm of peak SWE compared to the 300 mm reached by the ~~open loop reference~~ simulation.

5.3 Computational benchmarks

In a single cell benchmark experiment, the different data assimilation algorithms showed large variations in computational
855   cost (Figure 8). In this comparison ~~the values~~ with $N_e = 100$ particles the computational cost measured in wall clock time ranged from 39 (PBS) to 270 (EnKF-MDA, $N_a = 4$) seconds per cell per year ~~[s c^{-1} y^{-1}] in the case of the PBS to 270 [s c^{-1} y^{-1}] for 100 particles and 4 iterations in the case of the EnKF-MDA~~. These benchmarks showed that for all the cases the computational cost increased almost linearly with the number of particles. As expected, the iterative versions of the ensemble Kalman approaches are more demanding in terms of computational cost, due to the larger number of FSM2 simulations required
860   by the data assimilation algorithm. Filtering is also generally more expensive than (batch) smoothing since it invokes more frequent calls to I/O operations. This is especially clear for the costly iterative ensemble Kalman methods, but note that the relative increase in wall clock time for filtering relative to smoothing is more or less the same across all methods and corresponds to a factor of roughly 1.5.

6 Discussion

865   The results from the intercomparison of different data assimilation experiments exhibited a large variability in performance and computational cost across the schemes. This highlights the need for thorough testing, as the suitability of the assimilation algorithms will vary depending on the problem at hand. Despite the fact that ~~all~~ most of the data assimilation algorithms improved
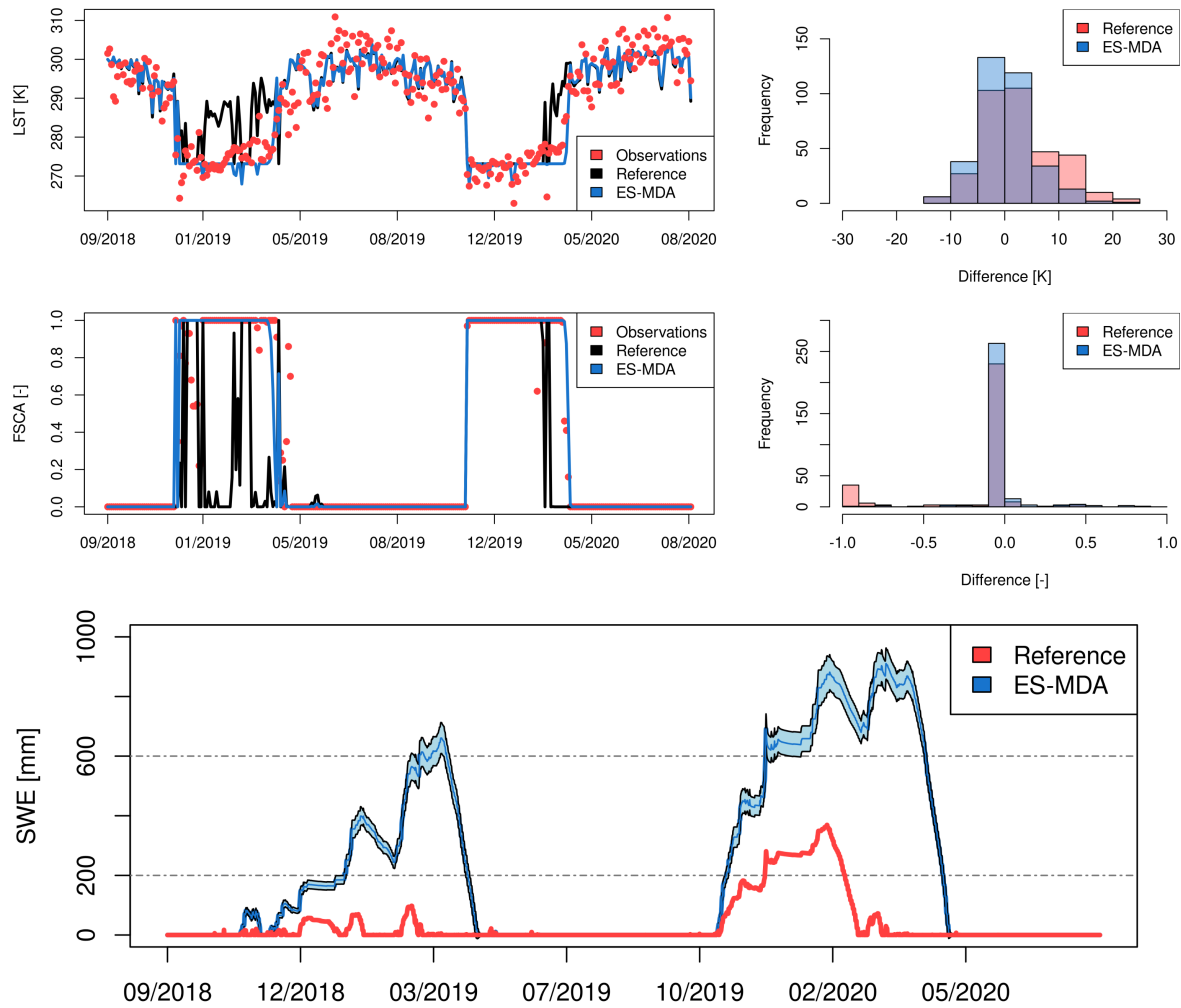
Figure 7. Joint ~~LSTA~~ LST (top panels) and FSCA (middle panels) data assimilation, and its influence on the SWE simulations (bottom panel).~~).~~

the simulations compared with the ~~deterministic open loop~~ reference simulations, their performance differed markedly. In fact, ~~this is sometimes not noticed~~ these differences tend not to be explored in the literature, where often the choice of one algorithm over others is not sufficiently ~~, or at all, justified~~ justified empirically. The lack of tools to compare the performance of different data assimilation algorithms has probably contributed to this problem, since it requires substantial coding effort to implement

870
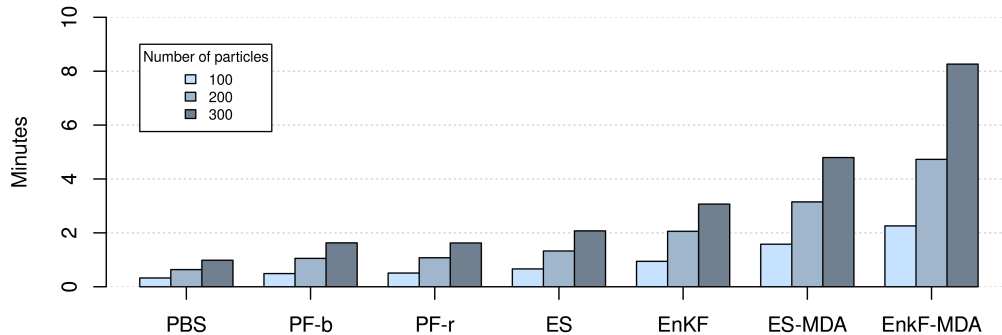
Figure 8. Benchmarks for the computational cost, expressed in terms of wall clock time, of the different assimilation schemes implemented in MuSA.

all available options in each data assimilation experiment. As an example, a case could be made that ensemble Kalman-based data assimilation approaches are often prematurely perceived as suboptimal for snow science applications due to the Gaussian linear assumption ~~(Helmert et al., 2018; Largeron et al., 2020)~~ (Helmert et al., 2018; Largeron et al., 2020). In spite of this ~~view~~,

875   our first experiment showed that the iterative version of ensemble-based Kalman smoothers outperformed the other smoother algorithms, as also found by ~~Aalstad et al. (2018). hese~~ Aalstad et al. (2018) and Pirk et al. (2022). These findings are consistent with the broader DA literature, where basic ~~particle methods~~ PFs (based on SIR) tend to suffer more from degeneracy due to the curse of dimensionality ~~(Snyder et al., 2008).~~ in high dimensional problems (Snyder et al., 2008). This degeneracy problem tends to be even worse with the PBS (based on SIS) due to the absence of resampling van Leeuwen and Evensen (1996); van Leeuwen (2009

880   .

In their review, van Leeuwen et al. (2019) suggest promising remedies to this problem with more sophisticated particle methods that invoke ~~remedies~~ innovations such as using ~~iterations or proposal distributions~~ better proposal distributions or iterations. The particle filter using redraws from a normal approximation to the posterior that we implemented in MuSA was loosely inspired by the use of proposal distributions and can overcome the degeneracy problems in the ~~more classic~~

885   classic bootstrap particle filter (see Table 1). The use of iterations was pursued by Stordal and Elsheikh (2015) who recast the ES-MDA in the framework of iterative particle methods, leading to improved performance on subsurface flow problems. This suggests that a hybridization of ensemble Kalman and particle methods (Papadakis et al., 2010; Pirk et al., 2022) may also be a promising avenue for future work. We reiterate that we are not advocating for one class of DA methods over another, the point we are making is rather that to do so prematurely would be unwise. Common wisdom embodied in the ´No Free Lunch'

890   theorems for optimization (Wolpert and Macready, 1997) warns us not to expect one particular algorithm to always perform

best, but rather that this will depend on the problem at hand. The power of MuSA is ~~that it~~ to provide a tool that simplifies the task of comparing different experimental set-ups by allowing the intercomparison of data assimilation algorithms as well as facilitating the implementation of new ones. In the future, ~~new~~ more data assimilation algorithms ~~will~~ could be implemented, including iterative versions of the PBS and particle filters as well MCMC methods which are the gold standard for Bayesian inference (Neal, 1993; Apte et al., 2007), but have received relatively ~~little~~ less attention from the snow community (Kolberg and Gottschalk, 2006) due to their often prohibitive computational cost.

The selection of the data assimilation algorithm is not the only possible source of variability in the updated simulations. As an example, MuSA has two different precipitation partitioning methods implemented. In addition to the fact that the selection of the precipitation phase partitioning method would have in the deterministic simulations, the perturbations in the forcing can also impact precipitation phase in different ways depending on the selected partitioning method. For example, perturbing the relative humidity forcing would only impact precipitation phase for one of the partitioning methods in FSM2. ~~In addition, different~~ Different strategies to generate the prior ensemble of simulations may markedly impact the performance of the data assimilation algorithms especially in the case of PBS and particle filters which are prone to degeneracy.

MuSA was also able to assimilate hyper-resolution snow depth maps in a distributed fashion. The assimilation of snow depth products has been shown to be a very robust approach to ~~SWE~~ snow depth estimation. The posterior maps of the perturbation parameters showed intricately detailed ~~and physically-consistent~~ spatial patterns, especially considering the fact that there is not any cell intercommunication in MuSA. The appearance of this ~~consistent~~ spatial pattern in the perturbation parameters indicates that they are compensating for a missing processes in the model, which in this case is most likely to be the wind-driven ablation and accumulation processes since most of the spatial variability in melt energy is provided by MicroMet shortwave radiation routine (e.g. Baba et al., 2019). Thus, MuSA can be used to study the importance of missing snow processes in the FSM2 model. In the particular case of the assimilation of hyper resolution snow depth maps, it may help to address the unresolved problem of wind redistribution in numerical snowpack modeling (Vionnet et al., 2021) by providing spatio-temporally continuous (i.e. gap free) and physically consistent reconstructions of the snowpack dynamics that can be used as a target when designing and calibrating new process parameterizations in snow model development.

Despite the fact that there are some other examples of assimilating snow depth products, most of these have been carried out ~~at the point scale~~ using in-situ observations (Cantet et al., 2019; Cluzet et al., 2022; Odry et al., 2022), or were developed at coarser spatiotemporal resolutions (Deschamps-Berger et al., 2022). To our knowledge the assimilation experiment performed in Section 5.1 is the first test of assimilating drone retrievals in a numerical snow model. This approach should be useful for the monitoring of hydrological experimental catchments in snow dominated areas, providing a cost effective method to generate spatially and temporally continuous reconstruction of the SWE distribution. Using MuSA in combination with the ever increasing capabilities of drones, it is possible to develop joint assimilation experiments using snow depth and other variables such as ~~snow surface temperature~~ LST or albedo at unprecedented resolutions. Furthermore, the assimilation of ~~hyper-resolution~~ high resolution snow depth maps may become a common practice in the future, even at wider scales, thanks to ~~the emerging remote sensing~~ satellite-based snow depth retrievals (Marti et al., 2016; Treichler and Kääb, 2017; Lievens et al., 2019).

925 The joint assimilation of FSCA and LST retrievals from MODIS ~~,~~ had a large impact in the updated SWE simulations. The assimilation of ~~the FSCA provides~~ LST has the potential to provide additional information when FSCA saturates at 1, ~~i.e.~~ for example during most of the ~~snow season in snow dominated areas, or~~ accumulation season and during the polar night in the absence of ~~light~~sunlight. This preliminary experiment suggests that the joint assimilation of LST and FSCA may be beneficial to improve distributed snow reanalyses at larger scales. There are several examples of assimilating SCA in both its

930 binary and fractional form (e.g. Margulis et al., 2016). However, the assimilation of LST in numerical snow models remains largely underexplored (Navari et al., 2016). This is probably due to the relative lack of accurate LST information at the desired scale for snow applications especially over complex terrain, as the currently available satellite sensors that are able to provide LST information either exhibit a spatial resolution that can be too coarse or revisit times that are too infrequent (Largeron et al., 2020). Nonetheless, given the agenda of several space agencies, LST information will be readily available in the future

935 at high spatiotemporal resolutions. This includes the Thermal infraRed Imaging Satellite for High-resolution Natural resource Assessment (TRISHNA; Lagouarde et al., 2018) ~~satellite from the Centre National d'Études Spatiales (CNES)~~ or the Land Surface Temperature Monitoring (LSTM) ~~mission from the European Space Agency (ESA)~~missions. Another logical next step to explore in terms of joint assimilation would be to assimilate FSCA and LST together with the aforementioned satellite-based snow depth retrievals (Marti et al., 2016; Treichler and Kääb, 2017; Lievens et al., 2019).

940 The computational cost of each data assimilation algorithm and implementation depends on ~~different~~ several factors. First, the computational cost of the Particle Filter and PBS approaches is different to the Kalman and the iterative Kalman approaches. This is due to the number of FSM2 runs required to run each algorithm. In the Particle Filter and PBS approaches, the number of FSM2 runs are $N_r = N_e$ where N_r denotes the number of runs (per grid cell) and N_e is the number of particles. On the other hand, the computational cost in terms of the number of FSM2 runs in the EnKF and ES will be $N_r = 2N_e$. For the iterative

945 versions of these ensemble Kalman-based approaches the number of runs will be $N_r = (N_a + 1)N_e$ where N_a is the number of assimilation cycles, i.e. the number of iterations in the ensemble Kalman update equations, selected by the user as explained in Section 3.5. In general, the ensemble Kalman techniques require more runs because they actually move (rather than just reweight) the parameter ensemble after assimilating the observations. To get the corresponding updated states (and predicted observations) FSM2 must be rerun with the updated parameters. In the non-iterative case this effectively requires a single rerun

950 of the FSM2 ensemble, while for the iterative case it requires N_a reruns of the FSM2 ensemble.

However, the number of FSM2 runs is not the only source of computational cost. The current version of MuSA is a wrapper around the FSM2 model. This simplifies the ~~implementation~~ inclusion of other snow models if required by the user. It also means that MuSA can easily incorporate FSM2 upgrades. Nonetheless, this implementation comes with a computational cost due to the I/O operations and the system calls that are done in the background when the MuSA system is running. The time ex-

955 pended on I/O operations and system calls will be higher in the parallel runs due to the use of disk space from different processes at the same time. This suggests that a ~~better encapsulation~~ more direct integration of FSM2 ~~running the snowpack simulations in memory may~~ that avoids I/O operations and system calls could improve the overall ~~MuSA performance~~ performance of MuSA.

It is worth noting that MuSA may be also used to implement simpler (and therefore) faster snowpack models (e.g. Aalstad et al., 2018), if numerical efficiency is required. Conversely, more sophisticated multilayer models such as Crocus (Vionnet et al., 2012) and SNOWPACK (Bartelt and Lehning, 2002) could also be encapsulated in MuSA in the future. The implementation of more sophisticated models that include detailed radiative transfer schemes may provide MuSA the capability of ingesting new remotely sensed information such as ~~surface reflectances~~ shortwave reflectances (Cluzet et al., 2020) or radar backscatter (Lievens et al., 2019). In addition, as FSM2 has support for different temporal resolutions, the future versions of MuSA will support different timesteps, allowing to reduce the computational cost both in terms of run time and data storage requirements. As such, there is room for reducing the computational cost of MuSA if necessary, which would open up the possibility of implementing it even at continental and hemispheric scales.

7 Conclusions

MuSA is a new snow data assimilation system that encapsulates the FSM2 snowpack model. There are 6 different ensemble-based data assimilation algorithms implemented in MuSA, as outlined in detail in Section 3, with ~~five~~ several different resampling strategies in the case of particle filters. ~~MuSA also supports direct insertion in the case of snow depth.~~ The data assimilation algorithms and the characteristics of the ensemble generation are provided by the user through simple configuration files. MuSA is able to assimilate different observational variables either independently or jointly (i.e. in the same assimilation step), even if these variables do not share the same time-step or if they have gaps. The system is highly scalable, such that it is possible to run it both on local machines and supercomputing infrastructures.

We used the MuSA system to assimilate snow depth maps derived from drone retrievals over the Izas experimental catchment for two different seasons at 5 meters spatial resolution. The behavior of each data assimilation algorithm differed considerably, demonstrating that case-specific testing of algorithms rather than just relying on the literature can be very helpful for designing successful snow data assimilation experiments. In addition, we developed a single cell data assimilation experiment, fusing LST and FSCA retrievals from MODIS with the FSM2 model. The results indicated a strong potential in the joint exploitation of these remotely sensed variables, suggesting that more research is needed on this regard. Finally, we presented a benchmark of the computing cost of the data assimilation algorithms. The choice of data assimilation algorithms had a considerable impact on the computational expense of the system, and should be considered for high resolution or large scale runs.

Code and data availability. The MuSA code is available at https://github.com/ealonsogzl/MuSA, the version of MuSA used for this paper (with a subset of input data) can be found at https://doi.org/10.5281/zenodo.7014570 while the complete input data for Izas used in the present study can be found at https://doi.org/10.5281/zenodo.7248635. The original FSM2 code is found at github.com/RichardEssery/FSM2, and in the MuSA repository with slight modifications from the original version. The MODIS data used herein, namely MOD10A1 (Hall et al., 2002) and MOD11A1 (Wan et al., 2015), are available for download from NSIDC and the NASA EOSDIS Land Processes DAAC, respectively. ERA5 data is available for download from the Copernicus Climate Data Store. MuSA can be launched on Linux/macOS or Windows platforms (via Windows Subsystem for Linux). MicroMet code is available upon request to G.E. Liston.

Author contributions. Conceptualization: EAG, KA, Data curation: EAG, MWB, JR, Formal analysis: EAG, KA, Funding acquisition: EAG, JLP, SG, Investigation: EAG, KA, Methodology: EAG, KA, Project administration: EAG, JLP, SG, Resources: EAG, JLP, Software: EAG, KA, Supervision: JLP, SG, Validation: EAG, KA, Visualization: EAG, KA, SG, Writing - original draft preparation: EAG, KA, Writing - review & editing: EAG, KA, MWB, JR, JLP, JF, RE, SG

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

Acknowledgements. Esteban Alonso-González has been funded by the CNES postdoctoral fellowship. Data from Izas experimental catchment was collected in the frame of HIDROIBERNIEVE- CGL2017-82216-R funded by the Spanish Ministry of Economy and Competitiveness project. Kristoffer Aalstad was funded by the Research Council of Norway through the Spot-On project (#301552), and acknowledges support from the LATICE strategic research area at the University of Oslo. J. Revuelto is supported by the Grant IJC2018-036260-I.

1000   References

Aalstad, K., Westermann, S., Schuler, T. V., Boike, J., and Bertino, L.: Ensemble-based assimilation of fractional snow-covered area satellite retrievals to estimate the snow distribution at Arctic sites, The Cryosphere, 12, 247–270, https://doi.org/10.5194/tc-12-247-2018, 2018.

Aalstad, K., Westermann, S., and Bertino, L.: Evaluating satellite retrieved fractional snow-covered area at a high-Arctic site using terrestrial photography, Remote Sensing of Environment, 239, 111 618, https://doi.org/10.1016/j.rse.2019.111618, 2020.

1005 Alonso-González, E., López-Moreno, J., Gascoin, S., García-Valdecasas Ojeda, M., Sanmiguel-Vallelado, A., Navarro-Serrano, F., Revuelto, J., Ceballos, A., Esteban-Parra, M., and Essery, R.: Daily gridded datasets of snow depth and snow water equivalent for the Iberian Peninsula from 1980 to 2014, Earth System Science Data, 10, 303–315, https://doi.org/10.5194/essd-10-303-2018, 2018.

Alonso-González, E., Gutmann, E., Aalstad, K., Fayad, A., Bouchet, M., and Gascoin, S.: Snowpack dynamics in the Lebanese mountains from quasi-dynamically downscaled ERA5 reanalysis updated by assimilating remotely sensed fractional snow-covered area, Hydrology

1010 and Earth System Sciences, 25, 4455–4471, https://doi.org/10.5194/hess-25-4455-2021, 2021.

Andreadis, K. M. and Lettenmaier, D. P.: Assimilating remotely sensed snow observations into a macroscale hydrology model, Advances in Water Resources, 29, 872–886, https://doi.org/10.1016/j.advwatres.2005.08.004, 2006.

Apte, A., Hairer, M., Stuart, A. M., and Voss, J.: Sampling the posterior: An approach to non-Gaussian data assimilation, Physica D: Nonlinear Phenomena, 230, 50–64, https://doi.org/10.1016/j.physd.2006.06.009, 2007.

1015 Baba, M., Gascoin, S., Kinnard, C., Marchane, A., and Hanich, L.: Effect of Digital Elevation Model Resolution on the Simulation of the Snow Cover Evolution in the High Atlas, 55(7), pp. 5360–5378, https://doi.org/10.1029/2018WR023789, 2019.

Baba, M. W., Gascoin, S., and Hanich, L.: Assimilation of Sentinel-2 data into a snowpack model in the High Atlas of Morocco, Remote Sensing, 10, 1982, https://doi.org/10.3390/rs10121982, 2018.

Baldo, E. and Margulis, S. A.: Assessment of a multiresolution snow reanalysis framework: a multidecadal reanalysis case over the upper

1020 Yampa River basin, Colorado, Hydrology and Earth System Sciences, 22, 3575–3587, https://doi.org/10.5194/hess-22-3575-2018, 2018.

Banner, K. M., Irvine, K. M., and Rodhouse, T. J.: The Use of Bayesian Priors in Ecology: The Good, the Bad and the Not Great, Methods in Ecology and Evolution, 11, 882–889, https://doi.org/10.1111/2041-210X.13407, 2020.

Bannister, R. N.: A review of operational methods of variational and ensemble-variational data assimilation, Quarterly Journal of the Royal Meteorological Society, 143, 607–633, https://doi.org/10.1002/qj.2982, 2017.

1025 Barnett, T. P., Adam, J. C., and Lettenmaier, D. P.: Potential impacts of a warming climate on water availability in snow-dominated regions, Nature, 438, 303–309, https://doi.org/10.1038/nature04141, 2005.

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, 2002.

Bauer, P., Thorpe, A., and Brunet, G.: The quiet revolution of numerical weather prediction, Nature, 525, 47–55,

1030 https://doi.org/10.1038/nature14956, 2015.

Bertino, L., Evensen, G., and Wackernagel, H.: Sequential Data Assimilation Techniques in Oceanography, International Statistical Review, 71, 223–241, https://doi.org/10.1111/j.1751-5823.2003.tb00194.x, 2003.

Bhardwaj, A., Singh, S., Sam, L., Bhardwaj, A., Martín-Torres, F. J., Singh, A., and Kumar, R.: MODIS-based estimates of strong snow surface temperature anomaly related to high altitude earthquakes of 2015, Remote Sensing of Environment, 188, 1–8,

1035 https://doi.org/10.1016/j.rse.2016.11.005, 2017.

Burgers, G., Jan van Leeuwen, P., and Evensen, G.: Analysis Scheme in the Ensemble Kalman Filter, Monthly Weather Review, 126, 1719–1724, https://doi.org/10.1175/1520-0493(1998)126<1719:ASITEK>2.0.CO;2, 1998.

Cantet, P., Boucher, M. A., Lachance-Coutier, S., Turcotte, R., and V. Fortin, V.: Using a Particle Filter to Estimate the Spatial Distribution of the Snowpack Water Equivalent, Journal of Hydrometeorology, https://doi.org/10.1175/JHM-D-18-0140.1, 2019.

1040 Cappé, O., Moulines, E., and Rydén, T.: Inference in Hidden Markov Models, Springer, https://doi.org/10.1007/0-387-28982-8, 2005.

Carrassi, A., Bocquet, M., Bertino, L., and Evensen, G.: Data assimilation in the geosciences: An overview of methods, issues, and perspectives, WIREs Climate Change, 9, e535, https://doi.org/10.1002/wcc.535, 2018.

Charrois, L., Cosme, E., Dumont, M., Lafaysse, M., Morin, S., Libois, Q., and Picard, G.: On the assimilation of optical reflectances and snow depth observations into a detailed snowpack model, The Cryosphere, 10, 1021–1038, https://doi.org/10.5194/tc-10-1021-2016, 2016.

1045 Chopin, N. and Papaspiliopoulos, O.: An Introduction to Sequential Monte Carlo, Springer, https://doi.org/10.1007/978-3-030-47845-2, 2020.

Clark, M. P., Slater, A. G., Barrett, A. P., Hay, L. E., McCabe, G. J., Rajagopalan, B., and Leavesley, G. H.: Assimilation of snow covered area information into hydrologic and land-surface models, Advances in Water Resources, 29, 1209–1221, https://doi.org/10.1016/j.advwatres.2005.10.001, 2006.

1050 Cluzet, B., Revuelto, J., Lafaysse, M., Tuzet, F., Cosme, E., Picard, G., et al.: Towards the assimilation of satellite reflectance into semi-distributed ensemble snowpack simulations, Cold Regions Science and Technology, 170, 102 918, https://doi.org/10.1016/j.coldregions.2019.102918, 2020.

Cluzet, B., Lafaysse, M., Cosme, E., Albergel, C., Meunier, L.-F., and Dumont, M.: CrocO_v1.0: a particle filter to assimilate snowpack observations in a spatialised framework, Geoscientific Model Development, 14, 1595–1614, https://doi.org/10.5194/gmd-14-1595-2021, 1055 2021.

Cluzet, B., Lafaysse, M., Deschamps-Berger, C., Vernay, M., and Dumont, M.: Propagating information from snow observations with CrocO ensemble data assimilation system: a 10-years case study over a snow depth observation network, The Cryosphere, https://doi.org/10.5194/tc-16-1281-2022, 2022.

Comola, F., Giometto, M. G., Salesky, S. T., Parlange, M. B., and Lehning, M.: Preferential Deposition of Snow and 1060 Dust Over Hills: Governing Processes and Relevant Scales, Journal of Geophysical Research: Atmospheres, 124, 7951–7974, https://doi.org/10.1029/2018JD029614, 2019.

Cortés, G. and Margulis, S.: Impacts of El Niño and La Niña on interannual snow accumulation in the Andes: Results from a high-resolution 31 year reanalysis: El Niño Effects on Andes Snow, Geophysical Research Letters, 44, 6859–6867, https://doi.org/10.1002/2017GL073826, 2017.

1065 Cosme, E., Verron, J., Brasseur, P., Blum, J., and Auroux, D.: Smoothing Problems in a Bayesian Framework and Their Linear Gaussian Solutions, Monthly Weather Review, 140, 683–695, https://doi.org/10.1175/MWR-D-10-05025.1, 2012.

De Lannoy, G. J. M., Reichle, R. H., Houser, P. R., Arsenault, K. R., Verhoest, N. E. C., and Pauwels, V. R. N.: Satellite-Scale Snow Water Equivalent Assimilation into a High-Resolution Land Surface Model, Journal of Hydrometeorology, 11, 352–369, https://doi.org/10.1175/2009JHM1192.1, 2010.

1070 De Lannoy, G. J. M., Reichle, R. H., Arsenault, K. R., Houser, P. R., Kumar, S., Verhoest, N. E. C., and Pauwels, V. R. N.: Multiscale assimilation of Advanced Microwave Scanning Radiometer-EOS snow water equivalent and Moderate Resolution Imaging Spectroradiometer snow cover fraction observations in northern Colorado, Water Resources Research, 48, W01 522, https://doi.org/10.1029/2011WR010588, 2012.

Dechant, C. and Moradkhani, H.: Radiance data assimilation for operational snow and streamflow forecasting, Advances in Water Resources, 34, 351–364, https://doi.org/10.1016/j.advwatres.2010.12.009, 2011.

Deschamps-Berger, C., Cluzet, B., Dumont, M., Lafaysse, M., Berthier, E., Fanise, P., and Gascoin, S.: Improving the Spatial Distribution of Snow Cover Simulations by Assimilation of Satellite Stereoscopic Imagery, Water Resources Research, 58, e2021WR030 271, https://doi.org/10.1029/2021WR030271, 2022.

Dozier, J., Bair, E. H., and Davis, R. E.: Estimating the spatial distribution of snow water equivalent in the world's mountains, Wiley Interdisciplinary Reviews: Water, 3, 461–474, https://doi.org/10.1002/wat2.1140, 2016.

Dumont, M., Durand, Y., Arnaud, Y., and Six, D.: Variational assimilation of albedo in a snowpack model and reconstruction of the spatial mass-balance distribution of an alpine glacier, Journal of Glaciology, 58, 151–164, https://doi.org/10.3189/2012JoG11J163, 2012.

Dunne, S. and Entekhabi, D.: An ensemble-based reanalysis approach to land data assimilation, Water Resources Research, 41, W02 013, https://doi.org/10.1029/2004WR003449, 2005.

Durand, M. and Margulis, S. A.: Feasibility Test of Multifrequency Radiometric Data Assimilation to Estimate Snow Water Equivalent, Journal of Hydrometeorology, 7, 443–457, https://doi.org/10.1175/JHM502.1, 2006.

Durand, M., Molotch, N. P., and Margulis, S. A.: Merging complementary remote sensing datasets in the context of snow water equivalent reconstruction, Remote Sensing of Environment, 112, 1212–1225, https://doi.org/10.1016/j.rse.2007.08.010, 2008.

Emerick, A. A. and Reynolds, A. C.: History matching time-lapse seismic data using the ensemble Kalman filter with multiple data assimilations, Computational Geosciences, 16, 639–539, https://doi.org/10.1007/s10596-012-9275-5, 2012.

Emerick, A. A. and Reynolds, A. C.: Ensemble smoother with multiple data assimilation, Computers & Geosciences, 55, 3–15, https://doi.org/10.1016/j.cageo.2012.03.011, 2013.

Essery, R.: A factorial snowpack model (FSM 1.0), Geoscientific Model Development, 8, 3867–3876, https://doi.org/10.5194/gmd-8-3867-2015, 2015.

Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, Journal of Geophysical Research, 99, 10 143, https://doi.org/10.1029/94JC00572, 1994.

Evensen, G.: Data Assimilation: The Ensemble Kalman Filter, Springer, https://doi.org/10.1007/978-3-642-03711-5, 2009.

Evensen, G.: Analysis of iterative ensemble smoothers for solving inverse problems, Computational Geosciences, 22, 885–908, https://doi.org/10.1007/s10596-018-9731-y, 2018.

Evensen, G.: Accounting for model errors in iterative ensemble smoothers, Computational Geosciences, 23, 761–775, https://doi.org/10.1007/s10596-019-9819-z, 2019.

Evensen, G., Vossepoel, F. C., and van Leeuwen, P. J.: Data Assimilation Fundamentals, Springer, https://doi.org/10.1007/978-3-030-96709-3, 2022.

Farchi, A. and Bocquet, M.: Review article: Comparison of local particle filters and new implementations, Nonlinear Processes in Geophysics, 25, 765–807, https://doi.org/10.5194/npg-25-765-2018, 2018.

Fayad, A. and Gascoin, S.: The role of liquid water percolation representation in estimating snow water equivalent in a Mediterranean mountain region (Mount Lebanon), Hydrology and Earth System Sciences, 24, 1527–1542, https://doi.org/10.5194/hess-24-1527-2020, 2020.

Fayad, A., Gascoin, S., Faour, G., López-Moreno, J. I., Drapeau, L., Page, M. L., and Escadafal, R.: Snow hydrology in Mediterranean mountain regions: A review, Journal of Hydrology, 551, 374–396, https://doi.org/10.1016/j.jhydrol.2017.05.063, 2017.

Fiddes, J. and Gruber, S.: TopoSCALE v.1.0: downscaling gridded climate data in complex terrain, Geoscientific Model Development, 7, 387–405, https://doi.org/10.5194/gmd-7-387-2014, 2014.

Fiddes, J., Aalstad, K., and Westermann, S.: Hyper-resolution ensemble-based snow reanalysis in mountain regions using clustering, Hydrology and Earth System Sciences, 23, 4717–4736, https://doi.org/10.5194/hess-23-4717-2019, 2019.

1115 Fiddes, J., Aalstad, K., and Lehning, M.: TopoCLIM: rapid topography-based downscaling of regional climate model output in complex terrain v1.1, Geoscientific Model Development, 15, 1753–1768, https://doi.org/10.5194/gmd-15-1753-2022, 2022.

Fletcher, S. J. and Zupanski, M.: A hybrid multivariate Normal and lognormal distributionfor data assimilation, Atmospheric Science Letters, https://doi.org/10.1002/asl.128, 2006.

Fowler, A. and van Leeuwen, P. J.: Observation impact in data assimilation: the effect of non-Gaussian observation error, Tellus A: Dynamic
1120 Meteorology and Oceanography, https://doi.org/10.3402/tellusa.v65i0.20035, 2013.

García-Ruiz, J. M., López-Moreno, J. I., Vicente-Serrano, S. M., Lasanta-Martínez, T., and Beguería, S.: Mediterranean water resources in a global change scenario, Earth-Science Reviews, 105, 121–139, https://doi.org/10.1016/j.earscirev.2011.01.006, 2011.

Gascoin, S., Grizonnet, M., Bouchet, M., Salgues, G., and Hagolle, O.: Theia Snow collection: High-resolution operational snow cover maps from Sentinel-2 and Landsat-8 data, Earth System Science Data, 11, 493–514, https://doi.org/10.5194/essd-11-493-2019, 2019.

1125 Gelaro, R., McCarty, W., Suárez, M. J., Todling, R., Molod, A., Takacs, L., et al.: The modern-era retrospective analysis for research and applications, version 2 (MERRA-2, Journal of Climate, 30, 5419–5454, https://doi.org/10.1175/JCLI-D-16-0758.1, 2017.

Gilks, W. R. and Berzuini, C.: Following a moving target-Monte Carlo inference for dynamic Bayesian models, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63, 127–146, https://doi.org/10.1111/1467-9868.00280, 2001.

Girotto, M., Cortés, G., Margulis, S. A., and Durand, M.: Examining spatial and temporal variability in snow water equivalent using a 27 year
1130 reanalysis: Kern River watershed, Sierra Nevada, Water Resources Research, 50, 6713–6734, https://doi.org/10.1002/2014WR015346, 2014a.

Girotto, M., Margulis, S. A., and Durand, M.: Probabilistic SWE reanalysis as a generalization of deterministic SWE reconstruction techniques, Hydrological Processes, 28, 3875–3895, https://doi.org/10.1002/hyp.9887, 2014b.

Girotto, M., Musselman, K. N., and Essery, R. L. H.: Data Assimilation Improves Estimates of Climate-Sensitive Seasonal Snow, Current
1135 Climate Change Reports, 6, 81–94, https://doi.org/10.1007/s40641-020-00159-7, 2020.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proceedings F Radar and Signal Processing, 140, 107, https://doi.org/10.1049/ip-f-2.1993.0015, 1993.

Günther, D., Marke, T., Essery, R., and Strasser, U.: Uncertainties in Snowpack Simulations—Assessing the Impact of Model Structure, Parameter Choice, and Forcing Data Error on Point-Scale Energy Balance Snow Model Performance, Water Resources Research, 55,
1140 2779–2800, https://doi.org/10.1029/2018WR023403, 2019.

Gutmann, E., Barstad, I., Clark, M., Arnold, J., and Rasmussen, R.: The Intermediate Complexity Atmospheric Research model (ICAR, Journal of Hydrometeorology, 17, 957–973, https://doi.org/10.1175/JHM-D-15-0155.1, 2016.

Hall, D. and Riggs, G.: MODIS/Terra Snow Cover Daily L3 Global 500m SIN Grid Version 6, NASA National Snow and Ice Data Center Distributed Active Archive Center, https://doi.org/10.5067/MODIS/MOD10A1.006, 2016.

1145 Hall, D. K., Riggs, G. A., Salomonson, V. V., DiGirolamo, N. E., and Bayr, K. J.: MODIS snow-cover products, Remote Sensing of Environment, 83, 181–194, https://doi.org/10.1016/S0034-4257(02)00095-0, 2002.

Harder, P. and Pomeroy, J.: Estimating precipitation phase using a psychrometric energy balance method, Hydrological Processes, 27, 1901–1914, https://doi.org/10.1002/hyp.9799, 2013.

Havens, S., Marks, D., Kormos, P., and Hedrick, A.: Spatial Modeling for Resources Framework (SMRF): A modular framework for developing spatial forcing data for snow modeling in mountain basins, Computers & Geosciences, 109, 295–304, https://doi.org/10.1016/j.cageo.2017.08.016, 2017.

Helmert, J., Şensoy Şorman, A., Alvarado Montero, R., De Michele, C., De Rosnay, P., Dumont, M., et al.: Review of Snow Data Assimilation Methods for Hydrological, Land Surface, Meteorological and Climate Models: Results from a COST HarmoSnow Survey, Geosciences, 8, 489, https://doi.org/10.3390/geosciences8120489, 2018.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., et al.: The ERA5 global reanalysis, Quarterly Journal of the Royal Meteorological Society, 146, 1999–2049, https://doi.org/10.1002/qj.3803, 2020.

Hou, J., Huang, C., Chen, W., and Zhang, Y.: Improving Snow Estimates Through Assimilation of MODIS Fractional Snow Cover Data Using Machine Learning Algorithms and the Common Land Model, Water Resources Research, 57, e2020WR029 010, https://doi.org/10.1029/2020WR029010, 2021.

Huang, C., Newman, A. J., Clark, M. P., Wood, A. W., and Zheng, X.: Evaluation of snow data assimilation using the ensemble Kalman filter for seasonal streamflow prediction in the western United States, Hydrology and Earth System Sciences, 21, 635–650, https://doi.org/10.5194/hess-21-635-2017, 2017.

Hüsler, F., Jonas, T., Riffler, M., Musial, J. P., and Wunderle, S.: A satellite-based snow cover climatology (1985-2011) for the European Alps derived from AVHRR data, The Cryosphere, 8, 73–90, https://doi.org/10.5194/tc-8-73-2014, 2014.

Jazwinski, A.: Stochastic Processes and Filtering Theory, Academic Press, 1970.

Katzfuss, M., Stroud, J. R., and Wikle, C. K.: Understanding the Ensemble Kalman Filter, The American Statistician, 70, 350–357, https://doi.org/10.1080/00031305.2016.1141709, 2016.

Kavetski, D. and Kuczera, G.: Model smoothing strategies to remove microscale discontinuities and spurious secondary optima in objective functions in hydrological calibration, Water Resources Research, 43, W03 411, https://doi.org/10.1029/2006WR005195, 2007.

Kokhanovsky, A., Box, J. E., Vandecrux, B., Mankoff, K. D., Lamare, M., Smirnov, A., and Kern, M.: The determination of snow albedo from satellite measurements using fast atmospheric correction technique, Remote Sensing, 12, 234, https://doi.org/10.3390/rs12020234, 2020.

Kolberg, S. A. and Gottschalk, L.: Updating of snow depletion curve with remote sensing data, Hydrological Processes, 20, 2363–2380, https://doi.org/10.1002/hyp.6060, 2006.

Lagouarde, J.-P., Bhattacharya, B. K., Crébassol, P., Gamet, P., Babu, S. S., Boulet, G., et al.: The Indian-French Trishna Mission: Earth Observation in the Thermal Infrared with High Spatio-Temporal Resolution, IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, pp. 4078–4081, https://doi.org/10.1109/IGARSS.2018.8518720, 2018.

Largeron, C., Dumont, M., Morin, S., Boone, A., Lafaysse, M., Metref, S., Cosme, E., Jonas, T., Wisntral, A., and Margulis, S.: Toward Snow Cover Estimation in Mountainous Areas Using Modern Data Assimilation Methods: A Review, Frontiers in Earth Science, 8, 325, https://doi.org/10.3389/feart.2020.00325, 2020.

Leisenring, M. and Moradkhani, H.: Snow water equivalent prediction using Bayesian data assimilation methods, Stochastic Environmental Research and Risk Assessment, 25, 253–270, https://doi.org/10.1007/s00477-010-0445-5, 2011.

Li, T., Bolic, M., and Djuric, P. M.: Resampling Methods for Particle Filtering: Classification, implementation, and strategies, IEEE Signal Processing Magazine, 32, 70–86, https://doi.org/10.1109/MSP.2014.2330626, 2015.

Lievens, H., Demuzere, M., Marshall, H. P., Reichle, R. H., Brucker, L., Brangers, I., et al.: Snow depth variability in the Northern Hemisphere mountains observed from space, Nature Communications, 10, 1–12, https://doi.org/10.1038/s41467-019-12566-y, 2019.

Lin, Y. and West, G.: Periconnection: A novel macroecological effect in snow cover phenology modulating ecosystem productivity over upper Northern Hemisphere, Science of the Total Environment, 805, 150 164, https://doi.org/10.1016/j.scitotenv.2021.150164, 2022.

Lindley, D.: The Philosophy of Statistics, Journal of the Royal Statistical Society. Series D (The Statistician), 2000.

1190   Liston, G. E. and Elder, K.: A Distributed Snow-Evolution Modeling System (SnowModel), Journal of Hydrometeorology, 7, 1259–1276, https://doi.org/10.1175/JHM548.1, 2006a.

Liston, G. E. and Elder, K.: A Meteorological Distribution System for High-Resolution Terrestrial Modeling (MicroMet), Journal of Hydrometeorology, 7, 217–234, https://doi.org/10.1175/JHM486.1, 2006b.

Liu, Y., Fang, Y., and Margulis, S. A.: Spatiotemporal distribution of seasonal snow water equivalent in High Mountain Asia from an 18-year

1195   Landsat–MODIS era snow reanalysis dataset, The Cryosphere, 15, 5261–5280, https://doi.org/10.5194/tc-15-5261-2021, 2021.

López-Moreno, J. I., Fassnacht, S. R., Beguería, S., and Latron, J. B. P.: Variability of snow depth at the plot scale: Implications for mean depth estimation and sampling strategies, The Cryosphere, 5, 617–629, https://doi.org/10.5194/tc-5-617-2011, 2011.

Luojus, K., Pulliainen, J., Takala, M., Lemmetyinen, J., Mortimer, C., Derksen, C., et al.: GlobSnow v3.0 Northern Hemisphere snow water equivalent dataset, Scientific Data, 8, 163, https://doi.org/10.1038/s41597-021-00939-2, 2021.

1200   MacKay, D. J. C.: Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.

Magnusson, J., Gustafsson, D., Hüsler, F., and Jonas, T.: Assimilation of point SWE data into a distributed snow cover model comparing two contrasting methods, Water Resources Research, 50, 7816–7835, https://doi.org/10.1002/2014WR015302, 2014.

Magnusson, J., Winstral, A., Stordal, A. S., Essery, R., and Jonas, T.: Improving physically based snow simulations by assimilating snow depths using the particle filter, Water Resources Research, 53, 1125–1143, https://doi.org/10.1002/2016WR019092, 2017.

1205   Margulis, S. A., Girotto, M., Cortés, G., and Durand, M.: A particle batch smoother approach to snow water equivalent estimation, Journal of Hydrometeorology, 16, 1752–1772, https://doi.org/10.1175/JHM-D-14-0177.1, 2015.

Margulis, S. A., Cortés, G., Girotto, M., and Durand, M.: A landsat-era Sierra Nevada snow reanalysis (1985-2015, Journal of Hydrometeorology, 17, 1203–1221, https://doi.org/10.1175/JHM-D-15-0177.1, 2016.

Margulis, S. A., Fang, Y., Li, D., Lettenmaier, D. P., and Andreadis, K.: The Utility of Infrequent Snow Depth Images for De-

1210   riving Continuous Space-Time Estimates of Seasonal Snow Water Equivalent, Geophysical Research Letters, 46, 5331–5340, https://doi.org/10.1029/2019GL082507, 2019.

Marti, R., Gascoin, S., Berthier, E., De Pinel, M., Houet, T., and Laffly, D.: Mapping snow depth in open alpine terrain from stereo satellite imagery, The Cryosphere, 10, 1361–1380, https://doi.org/10.5194/tc-10-1361-2016, 2016.

Molotch, N. P. and Bales, R. C.: SNOTEL representativeness in the Rio Grande headwaters on the basis of physiographics and remotely

1215   sensed snow cover persistence, Hydrological Processes, 20, 723–739, https://doi.org/10.1002/hyp.6128, 2006.

Murphy, K. P.: Probabilistic Machine Learning: An Introduction, MIT Press, probml.ai, 2022.

Navari, M., Margulis, S. A., Bateni, S. M., Tedesco, M., Alexander, P., and Fettweis, X.: Feasibility of improving a priori regional climate model estimates of Greenland ice sheet surface mass loss through assimilation of measured ice surface temperatures, The Cryosphere, 10, 103–120, https://doi.org/10.5194/tc-10-103-2016, 2016.

1220   Neal, R. M.: Probabilistic Inference Using Markov Chain Monte Carlo Methods, Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

Nearing, G., Tian, Y., Gupta, H. V., Clark, M., Harrison, K., and Weijs, S.: A philosophical basis for hydrological uncertainty, Hydrological Sciences Journal, https://doi.org/10.1080/02626667.2016.1183009, 2016.

Oaida, C. M., Reager, J. T., Andreadis, K. M., David, C. H., Levoe, S. R., Painter, T. H., et al.: A High-Resolution Data Assimilation Framework for Snow Water Equivalent Estimation across the Western United States and Validation with the Airborne Snow Observatory, Journal of Hydrometeorology, 20, 357–378, https://doi.org/10.1175/JHM-D-18-0009.1, 2019.

Odry, J., Boucher, M.-A., Lachance-Cloutier, S., Turcotte, R., and St-Louis, P.-Y.: Large-scale snow data assimilation using a spatialized particle filter: recovering the spatial structure of the particles, The Cryosphere, https://doi.org/10.5194/tc-16-3489-2022, 2022.

Palmer, T. N.: Stochastic weather and climate models, Nature Reviews Physics, 1, 463–471, https://doi.org/10.1038/s42254-019-0062-2, 2019.

Papadakis, N., Mémin, E., Cuzol, A., and Gengembre, N.: Data Assimilation with the Weighted Ensemble Kalman Filter, Tellus A: Dynamic Meteorology and Oceanography, 62, 673–697, https://doi.org/10.1111/j.1600-0870.2010.00461.x, 2010.

Piazzi, G., Thirel, G., Campo, L., and Gabellani, S.: A particle filter scheme for multivariate data assimilation into a point-scale snowpack model in an Alpine environment, The Cryosphere, 12, 2287–2306, https://doi.org/10.5194/tc-12-2287-2018, 2018.

Pirk, N., Aalstad, K., Westermann, S., Vatne, A., van Hove, A., Tallaksen, L., Cassiani, M., and Katul, G.: Inferring surface energy fluxes using drone data assimilation in large eddy simulations, Atmospheric Measurement Techniques Discussions, https://doi.org/10.5194/amt-2022-219, 2022.

Raleigh, M. S., Lundquist, J. D., and Clark, M. P.: Exploring the impact of forcing error characteristics on physically based snow simulations within a global sensitivity analysis framework, Hydrology and Earth System Sciences, 19, 3153–3179, https://doi.org/10.5194/hess-19-3153-2015, 2015.

Reich, S. and Cotter, C.: Probabilistic Forecasting and Bayesian Data Assimilation, Cambridge University Press, https://doi.org/10.1017/CBO9781107706804, 2015.

Revuelto, J., Azorin-Molina, C., Alonso-González, E., Sanmiguel-Vallelado, A., Navarro-Serrano, F., Rico, I., and Ignacio López-Moreno, J.: Meteorological and snow distribution data in the Izas Experimental Catchment (Spanish Pyrenees) from 2011 to 2017, Earth System Science Data, 9, 993–1005, https://doi.org/10.5194/essd-9-993-2017, 2017.

Revuelto, J., Alonso-Gonzalez, E., Vidaller-Gayan, I., Lacroix, E., Izagirre, E., Rodríguez-López, G., and López-Moreno, J.: Intercomparison of UAV platforms for mapping snow depth distribution in complex alpine terrain, Cold Regions Science and Technology, 190, 103 344, https://doi.org/10.1016/j.coldregions.2021.103344, 2021a.

Revuelto, J., Cluzet, B., Duran, N., Fructus, M., Lafaysse, M., Cosme, E., and Dumont, M.: Assimilation of surface reflectance in snow simulations: Impact on bulk snow variables, Journal of Hydrology, 603, 126 966, https://doi.org/10.1016/j.jhydrol.2021.126966, 2021b.

Rew, R. and Davis, G.: NetCDF: an interface for scientific data access, IEEE Computer Graphics and Applications, 10, 76–82, https://doi.org/10.1109/38.56302, 1990.

Särkkä, S.: Bayesian Filtering and Smoothing, Cambridge University Press, https://doi.org/10.1017/CBO9781139344203, 2013.

Slater, A. and Clark, M.: Snow Data Assimilation via an Ensemble Kalman Filter, Journal of Hydrometeorology, 7, 478–493, https://doi.org/10.1175/JHM505.1, 2006.

Smith, A. F. M. and Gelfand, A. E.: Bayesian Statistics without Tears: A Sampling–Resampling Perspective, The American Statistician, 46, 84–88, https://doi.org/10.1080/00031305.1992.10475856, 1992.

Smyth, E. J., Raleigh, M. S., and Small, E. E.: Particle Filter Data Assimilation of Monthly Snow Depth Observations Improves Estimation of Snow Density and SWE, Water Resources Research, 55, 1296–1311, https://doi.org/10.1029/2018WR023400, 2019.

Smyth, E. J., Raleigh, M. S., and Small, E. E.: Improving SWE Estimation With Data Assimilation: The Influence of Snow Depth Observation Timing and Uncertainty, Water Resources Research, 56, e2019WR026 853, https://doi.org/10.1029/2019WR026853, 2020.

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J.: Obstacles to High-Dimensional Particle Filtering, Monthly Weather Review, 136, 4629–4640, https://doi.org/10.1175/2008MWR2529.1, 2008.

Stigter, E. E., Wanders, N., Saloranta, T. M., Shea, J. M., Bierkens, M. F. P., and Immerzeel, W. W.: Assimilation of snow cover and snow depth into a snow model to estimate snow water equivalent and snowmelt runoff in a Himalayan catchment, The Cryosphere, 11, 1647–1664, https://doi.org/10.5194/tc-11-1647-2017, 2017.

Stordal, A. S. and Elsheikh, A. H.: Iterative ensemble smoothers in the annealed importance sampling framework, Advances in Water Resources, 86, 231–239, https://doi.org/10.1016/j.advwatres.2015.09.030, 2015.

Sturm, M., Goldstein, M. A., and Parr, C.: Water and life from snow: A trillion dollar science question, Water Resources Research, 53, 3534–3544, https://doi.org/10.1002/2017WR020840, 2017.

Thirel, G., Salamon, P., Burek, P., and Kalas, M.: Assimilation of MODIS Snow Cover Area Data in a Distributed Hydrological Model Using the Particle Filter, Remote Sensing, 5, 5825–5850, https://doi.org/10.3390/rs5115825, 2013.

Treichler, D. and Kääb, A.: Snow depth from ICESat laser altimetry — A test study in southern Norway, Remote Sensing of Environment, 191, 389–401, https://doi.org/10.1016/j.rse.2017.01.022, 2017.

van Leeuwen, P. J.: Particle Filtering in Geophysical Systems, Monthly Weather Review, https://doi.org/10.1175/2009MWR2835.1, 2009.

van Leeuwen, P. J.: A consistent interpretation of the stochastic version of the Ensemble Kalman Filter, Quarterly Journal of the Royal Meteorological Society, 146, 2815–2825, https://doi.org/10.1002/qj.3819, 2020.

van Leeuwen, P. J. and Evensen, G.: Data Assimilation and Inverse Methods in Terms of a Probabilistic Formulation, Monthly Weather Review, 124, 2898–2913, https://doi.org/10.1175/1520-0493(1996)124<2898:DAAIMI>2.0.CO;2, 1996.

van Leeuwen, P. J., Künsch, H. R., Nerger, L., Potthast, R., and Reich, S.: Particle filters for high-dimensional geoscience applications: A review, Quarterly Journal of the Royal Meteorological Society, 145, 2335–2365, https://doi.org/10.1002/qj.3551, 2019.

Vionnet, V., Brun, E., Morin, S., Boone, A., Faroux, S., Le Moigne, P., et al.: The detailed snowpack scheme Crocus and its implementation in SURFEX v7, Geoscientific Model Development, 5, 773–791, https://doi.org/10.5194/gmd-5-773-2012, 2012.

Vionnet, V., Marsh, C. B., Menounos, B., Gascoin, S., Wayand, N. E., Shea, J., et al.: Multi-scale snowdrift-permitting modelling of mountain snowpack, The Cryosphere, 15, 743–769, https://doi.org/10.5194/tc-15-743-2021, 2021.

Wan, Z., Hook, S., and Hulley, G.: MOD11A1 MODIS/Terra Land Surface Temperature/Emissivity Daily L3 Global 1km SIN Grid V006, NASA EOSDIS Land Processes DAAC, https://doi.org/10.5067/MODIS/MOD11A1.006, 2015.

Westermann, S. et al.: The CryoGrid community model (version 1.0) – a multi-physics toolbox for climate-driven simulations in the terrestrial cryosphere, Geoscientific Model Development Discussions, https://doi.org/10.5194/gmd-2022-127, 2022.

Wikle, C. K. and Berliner, L. M.: A Bayesian tutorial for data assimilation, Physica D: Nonlinear Phenomena, 230, 1–16, https://doi.org/10.1016/j.physd.2006.09.017, 2007.

Wolpert, D. and Macready, W.: No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation, https://doi.org/10.1109/4235.585893, 1997.

Wrzesien, M. L., Durand, M. T., Pavelsky, T. M., Howat, I. M., Margulis, S. A., and Huning, L. S.: Comparison of methods to estimate snow water equivalent at the mountain range scale: A case study of the California Sierra Nevada, Journal of Hydrometeorology, 18, 1101–1119, https://doi.org/10.1175/JHM-D-16-0246.1, 2017.

Wrzesien, M. L., Durand, M. T., Pavelsky, T. M., Kapnick, S. B., Zhang, Y., Guo, J., and Shum, C. K.: A New Estimate of North American Mountain Snow Accumulation From Regional Climate Model Simulations, Geophysical Research Letters, 45, 1423–1432, https://doi.org/10.1002/2017GL076664, 2018.