Neglecting model parametric uncertainty can drastically underestimate flood risks

Sanjib Sharma\textsuperscript{1,1,1}, Benjamin Seiyon Lee\textsuperscript{2,2,2}, Iman Hosseini-Shakib\textsuperscript{1,1,1}, Murali Haran\textsuperscript{1,1,1}, and Klaus Keller\textsuperscript{3,3,3}

\textsuperscript{1}Pennsylvania State University
\textsuperscript{2}George Mason University
\textsuperscript{3}Thayer School of Engineering at Dartmouth College

December 7, 2022

Abstract

Floods drive dynamic and deeply uncertain risks for people and infrastructures. Uncertainty characterization is a crucial step in improving the predictive understanding of multi-sector dynamics and the design of risk-management strategies. Current approaches to estimate flood hazards often sample only a relatively small subset of the known unknowns, for example the uncertainties surrounding the model parameters. This approach neglects the impacts of key uncertainties on hazards and system dynamics. Here we mainstream a recently developed method for Bayesian inference to calibrate a computationally expensive distributed hydrologic model. We compare three different calibration approaches: (1) stepwise line search, (2) precalibration or screening, and (3) the new Fast Model Calibrations (FaMoS) approach. FaMoS deploys a particle-based approach that takes advantage of the massive parallelization afforded by modern high-performance computing systems. We quantify how neglecting parametric uncertainty and data discrepancy can drastically underestimate extreme flood events and risks. Precalibration improves prediction skill score over a stepwise line search. The Bayesian calibration improves the uncertainty characterization of model parameters and flood risk projections.
Neglecting model parametric uncertainty can drastically underestimate flood risks

Sanjib Sharma¹, Benjamin Seiyon Lee², Iman Hosseini-Shakib¹, Murali Haran³, and Klaus Keller⁴

¹Earth and Environmental Systems Institute, Pennsylvania State University, University Park, PA, USA
²Department of Statistics, George Mason University, Fairfax, VA, USA
³Department of Statistics, Pennsylvania State University, University Park, PA, USA
⁴Thayer School of Engineering at Dartmouth College, Hanover, NH, USA

Corresponding author: Sanjib Sharma (svs6308@psu.edu)

Key points:
• We implement a sequential Monte Carlo particle-based Fast Model Calibrations (FaMoS) approach to improve the characterization of distributed hydrologic model parameters.
• FaMoS demonstrates a relatively higher prediction skill than stepwise line search and precalibration.
• Accounting for model parametric uncertainty and observation errors can improve the uncertainty characterization of flood-damage projections.

Abstract
Floods drive dynamic and deeply uncertain risks for people and infrastructures. Uncertainty characterization is a crucial step in improving the predictive understanding of multi-sector dynamics and the design of risk-management strategies. Current approaches to estimate flood hazards often sample only a relatively small subset of the known unknowns, for example the uncertainties surrounding the model parameters. This approach neglects the impacts of key uncertainties on hazards and system dynamics. Here we mainstream a recently developed method for Bayesian inference to calibrate a computationally expensive distributed hydrologic model. We compare three different calibration approaches: (1) stepwise line search, (2) precalibration or screening, and (3) the Fast Model Calibrations (FaMoS) approach. FaMoS deploys a particle-based approach that takes advantage of the massive parallelization afforded by modern high-performance computing systems. We quantify how neglecting parametric uncertainty and data discrepancy can drastically underestimate extreme flood events and risks. Precalibration improves prediction skill score over a stepwise line search. The Bayesian calibration improves the uncertainty characterization of model parameters and flood risk projections.
1. Motivation and Introduction

Floods pose major risks to people and property (Alfieri et al., 2017; Wing et al., 2018; Winsemius et al., 2015). These risks are dynamic and deeply uncertain (Merz et al., 2010; Read & Vogel, 2015; Ruckert et al., 2019; Zarekarizi et al., 2020). It is important to characterize the uncertainties surrounding flood hazards in order to understand the impacts on multi-sector dynamics and to inform the design of risk-management strategies (Boulange et al., 2021; Chester et al., 2020; Liu & Merwade, 2018; Salas et al., 2018b; Wasko et al., 2021; Wong & Keller, 2017).

Hydrologic models are commonly used to understand hydrological processes, predict the response of hydrological systems to changing stresses, and provide boundary conditions to estimate flood hazards and risks (Bates et al., 2021; Brunner et al., 2020; Judi et al., 2018; Koren et al., 2004; Rajib et al., 2020; Thorstensen et al., 2016). However, hydrologic projections are subject to uncertainties such as from model structures, parameters and forcings (Gupta et al., 2012; Kavetski et al., 2006; Beven, 2014; Fisher & Koven, 2020; Hu et al., 2019; Mendoza et al., 2015). Parametric uncertainty can arise, for example, from the epistemic uncertainties about model parameters (Vrugt et al., 2003), the associated prior distributions (Tang et al., 2016), spatial-resolutions and objective functions (Melsen et al., 2019), and different choices of calibration approaches (Kavetski et al., 2018). Hydrologic models need to resolve the complex response of multiple processes (e.g., land surface characteristics, soil properties and climate variability) with strong nonlinear interactions and often few observations. Characterizing parametric uncertainty can be critical to improve prediction credibility and inform decision-making, for example, in the context of water-resources planning and flood-risk management (Herman et al., 2013; Ruckert et al., 2019; Wong & Keller, 2017; Zarekarizi et al., 2020).

Previous studies provide valuable new insights on flood hazard and risk estimates using model simulations (Bates et al., 2021; Judi et al., 2018; Rajib et al., 2020; Sanders et al., 2020; Sharma et al., 2021; Wing et al., 2018). For example, Judi et al. (2018) demonstrates an integrated multimodel multiscale simulation approach to evaluate social, economic, and infrastructure resilience to future flooding. Rajib et al. (2020) develops a coupled land surface hydrologic and river hydraulic modeling framework to provide regional flood hazard and risk estimates. Bates et al. (2021) presents estimates of current and future flood risk for all properties in the conterminous United States using a combined modeling approach considering river, coastal, or rainfall flooding. These studies typically obtain an optimal parameter set that produces the best possible agreement
between simulated and observed streamflow hydrographs at target locations. These previous studies break important new ground, but are mostly silent on the impacts of parametric uncertainties on hazards and dynamics. Neglecting parametric uncertainties can underestimate the tails of flood hazard probability distribution (Bates et al., 2021; Mendoza et al., 2015; Rojas et al., 2020; Salas et al., 2018a), and can result in poor decisions and outcomes (Ruckert et al., 2019; Wong & Keller, 2017; Zarekarizi et al., 2020).

Several studies on hydrologic model calibration have focused on manually adjusting a subset of model parameters (Bitew & Gebremichael, 2011; Siddique & Mejia, 2017). These manual calibrations typically rely on visual inspection of streamflow hydrograph and a trial and error-based procedure; hence, this method can be rather labor-intensive and time-consuming (Lahmers et al., 2021; Siddique & Mejia, 2017). A conceptual intuitive and relatively simple to implement approach for uncertainty characterization is the Generalized Likelihood Uncertainty Estimation (GLUE) method (Beven and Binley, 1992). The GLUE method has many advantages and can provide very useful insights, but several studies point to potential improvements with regard to subjective decisions on the likelihood function and implementing a statistically consistent error model (Blasone et al., 2008; Stedinger et al., 2008). A more complex approach adopted in this area is automatic parameter optimization (Kamali et al., 2013; Van Liew et al., 2005). Automatic calibration relies on systematic search approaches to find the best parameter values based on predefined single- and/or multi-objective functions (Kamali et al., 2013). Some studies use surrogate methods such as Gaussian process-based emulators to help identify best-fit parameters (Gou et al., 2020; Pianosi et al., 2016; Razavi & Tolson, 2013). Gou et al. (2020) presents an automatic calibration framework that combines sensitivity analysis and surrogate-based optimization for calibrating catchment-specific hydrologic model parameters. Surrogate-based methods are typically limited to cases with relatively fewer model parameters because training a surrogate model can be computationally prohibitive with high-dimensional inputs due to the large number of training data required (Hwang & Martins, 2018; Lee et al., 2020; Liu & Guillas, 2017) or repeated evaluations of the gradient of the model output with respect to the input parameters (Constantine et al., 2014; Lataniotis et al., 2020).

Bayesian calibration of hydrologic models have become increasingly popular (Hsu et al., 2009; Jeremiah et al., 2011; Kavetski et al., 2018; Raje & Krishnan, 2012; Razavi & Tolson, 2013; Shafii et al., 2015; Su et al., 2018; Zhu et al., 2018). For example, Vrugt et al., (2008) employ an
adaptive Metropolis Markov chain Monte Carlo (MCMC) sampling scheme-Differential Evolution Adaptive Metropolis (DREAM) algorithm to explore the entire parameter space of a hydrologic model. Different variants of DREAM algorithm (Vrugt et al., 2008; Vrugt et al., 2009; Laloy and Vrugt, 2012) demonstrate the value of Bayesian approaches on model calibration. Jeremiah et al. (2011) demonstrate an improved efficiency of Sequential Monte Carlo approach over the Adaptive Metropolis MCMC samplers in exploring the parameter space where the optimal solutions lie in the tails of the prescribed prior distribution. Su et al. (2018) uses a Bayesian hierarchical model to calibrate the Priestly–Taylor Jet Propulsion Laboratory model using observed evapotranspiration measurements. Given the relatively short model run times, the hierarchical model can be fit using the Differential Evolution Markov Chain (Braak, 2006; Storn & Price, 1997), a population MCMC algorithm. Zhu et al. (2018) calibrates eight parameters of a conceptual water balance model using a Particle Evolution Metropolis Sequential Monte Carlo (PEM-SMC). The PEM-SMC algorithm evaluates the water balance model 2,000 times sequentially, which may be computationally prohibitive for distributed hydrologic models with longer run times. These studies break important new ground, but focus on calibrating (1) average response of process over the watershed using a lumped hydrological model; (2) limited number of model parameters; (3) low-to-moderate flow threshold; and (4) relatively small basins. However, the computational requirement can be drastically larger for fully distributed hydrological modeling over the large basin and with a large number of sensitive parameters.

Here we expand on previous studies and demonstrate an implementation of a Bayesian model calibration framework by: (1) considering a computationally expensive distributed hydrologic model; (2) taking advantage of the massive parallelization afforded by modern high-performance computing systems; (3) focusing on a large number of extreme streamflow events; (4) characterizing model parametric uncertainty, and (5) assessing the impacts of uncertainty characterization on projected flood-hazards and -risks.

2. Bayesian Model Calibration

Various algorithms exist for characterizing hydrologic model parametric uncertainty, including the multicriteria approach (Gupta et al., 1998), Generalized likelihood Uncertainty Estimation (GLUE) (Beven and Binley, 1992), Shuffled Complex Evolution Metropolis algorithm (SCEM-UA) (Duan et al., 1992; Sorooshian et al., 1993), Shuffled Complex Evolution Metropolis
Bayesian computer model calibration (Bayarri et al., 2007a; Higdon et al., 2004; Kennedy & O’Hagan, 2001; Sacks et al., 1989) typically addresses several (potentially overlapping) objectives: (1) estimate the input parameters (in other words: what is the best parameter estimates); (2) quantify the parametric uncertainty (in other words: what is the joint probability density function of the parameters); and (3) infer the parameters of the observational error model and discrepancy terms. These parameter estimates are impacted by factors such as model-observation discrepancy (Bayarri et al., 2007b; Brynjarsdóttir & O’Hagan, 2014; Kennedy & O’Hagan, 2001) and measurement errors. The Bayesian model calibration framework (see the discussion in Kennedy and O’Hagan, 2001) facilitates both parameter estimation and uncertainty quantification while also accounting for external sources of uncertainty (e.g., discrepancy and measurement errors). For each model parameter, we specify prior distributions based on expert knowledge and then update the priors by comparing the model runs to the observed data. The update proceeds by placing more weight on the parameter sets whose corresponding model runs align better with the observations. The resulting posterior (updated) distribution naturally provides both point and interval estimates of the model parameters in light of the newly acquired data. Let $\theta$ be the vector of the model parameters, $\sigma^2$ the variance of the (assumed) independent and identically distributed observational error, and $\delta$ the discrepancy term. The posterior distribution $\tilde{\pi}(\theta, \sigma^2, \delta | Z)$ is defined as:

$$
\tilde{\pi}(\theta, \sigma^2, \delta | Z) \propto L(\theta, \sigma^2, \delta | Z) \times \pi(\theta) \times \pi(\sigma^2) \times \pi(\delta),
$$

where $\tilde{\pi}(\theta, \sigma^2, \delta | Z)$ and $\pi(\cdot)$ denotes the probability density function of the posterior and prior distributions, respectively. $L(\theta, \sigma^2, \delta | Z)$ is the likelihood function based upon the hydrological model output, discrepancy term, and the observational error model (see Appendix).

For complex deterministic models, the posterior distribution may not be available in closed form (Higdon, 2003; Oakley, 2009). In this case, a common approach is to approximate the posterior via sampling approaches such as Markov chain Monte Carlo (MCMC) or Sequential Monte Carlo. The choice of sampling approaches in influenced by several factors including: (1) the computational time requirements for a single model evaluation; (2) the number of model parameters to be calibrated, (3) the degree to which the algorithm can be parallelized, (4) the available computation environment, and (5) the available time for the computations. Markov chain
Monte Carlo methods with the true model can be an excellent choice for models with short single model run times (Asher et al., 2015; Gramacy, 2020; Lee et al., 2020). Surrogate modeling (i.e. emulation-calibration) approaches replace the hydrologic model with a faster surrogate model within the calibration framework; however, constructing a high-fidelity surrogate model may be computationally prohibitive for high-dimensional input spaces (X. Liu and Guillas 2017; Gramacy 2020). Sequential Monte Carlo (SMC) (Lee et al. 2020; Kalyanaraman et al. 2016; Papaioannou, Papadimitriou, and Straub 2016; Kantas, Beskos, and Jasra 2014; Morzfeld et al. 2018) methods can be a practical alternative for calibrating hydrological models with a larger number of input parameters.

2.1. The Fast Model Calibrations (FaMoS) approach

We use a sequential Monte Carlo particle-based approach that relies on massive parallelization afforded by a high-performance computing system to efficiently calibrate a distributed hydrologic model in a relatively large watershed with a number of extreme events.

Fast Model Calibrations (FaMoS) approach (Lee et al., 2020) provides an approximation of the posterior distribution by (i) generating an adaptive posterior incorporation schedule to preserve particle diversity; (ii) requiring very few Metropolis-Hastings updates in the mutation stages; and (iii) lending itself to parallel operations distributed across thousands of processors. We provide technical details about FaMoS in the Appendix.

FaMoS approximates the posterior distribution of the model parameters using a series of sampling, reweighting, and re-sampling steps. The basic premise of sampling-importance resampling (Gordon et al., 1993) is to draw independent samples from the model parameters’ prior distribution and retain the parameter sets whose corresponding outputs closely resemble the actual observations. Each parameter set is then assigned weights, which are proportional to the likelihood function $L(\theta|Z)$. The parameter sets whose model outputs fit the observed data well are given larger weights and those that do not are assigned smaller weights. The (importance) weights $w(\theta)$ are defined as:

$$w(\theta) = \frac{f(\theta)}{q(\theta)} = \frac{\pi(\theta|Z)}{\pi(\theta)},$$

where $f(\theta)$ is the target function and $q(\theta)$ is the importance function. In this context, we specify the target function as the posterior distribution of the model parameters $\tilde{\pi}(\theta|Z)$ and importance
function as the prior distribution of the parameters $\pi(\theta)$. We approximate $\hat{\pi}(\theta | Z)$ using the weighted empirical distribution $\hat{\pi}(\theta | Z)$ defined as:

$$\hat{\pi}(\theta | Z) \approx \hat{\pi}(\theta | Z) = \sum_{i=1}^{N} w(\theta^{(i)}) \delta(\theta^{(i)}) ,$$  \hspace{1cm} (2)

where $w(\theta^{(i)})$ is the importance weight and $\delta(\theta^{(i)})$ is a Dirac measure at $\theta^{(i)}$ for the $i$-th sample.

In the fast particle-based approach (Lee et al. 2020), we draw an initial ensemble of model parameters (particles) from the prior distribution (i.e., importance function) and approximate the posterior distribution (target function) using the initial ensemble. When there is very little overlap in the high-probability regions of the prior and posterior distribution, the initial ensemble may not adequately approximate the posterior distribution due to: (1) weight degeneracy, where the vast majority of particles have near-zero weights; and (2) sample impoverishment, where we “resample” the existing particles based on the weights, and we are left with multiple copies of a few unique particles. When there is very little overlap in the high-probability regions of the prior and posterior distribution, the initial ensemble may not adequately approximate the posterior distribution due to: (1) weight degeneracy, where the vast majority of particles have near-zero weights; and (2) sample impoverishment. Sample impoverishment occurs when we are left with multiple copies of a few unique particles after a “resampling” stage. In FaMoS, the resulting particles are “resampled” through multinomial sampling based on the importance weights $w(\theta i)$ then “mutated” or “jittered” using Metropolis-Hastings updates. Please see the Appendix for additional details.

FaMoS (Lee et al, 2020) mitigates these issues by gradually building up to the posterior distribution, a technique from iterated batch importance sampling (Chopin, 2002) and Sequential Monte Carlo. Here, we consider a series of intermediate posterior distributions where those earlier in the series closely resemble the prior distribution and those at the latter part better resemble the full posterior distribution. In the first cycle, we use particles from the prior distribution to approximate an earlier intermediate posterior distribution. In the subsequent cycles, we use samples from an intermediate posterior distribution to approximate a later intermediate posterior distribution. We end the algorithm when the target distribution is the final posterior distribution.

For cycles $t=1,...,T$, the $t$-th intermediate posterior distribution is:

$$\tilde{\pi}_t(\theta | Z) \propto L(|Z|)^{\gamma_t} \times \pi(\theta), \hspace{1cm} (3)$$

where $\gamma_t$ denotes the incorporation factor such that $0 = \gamma_0 \leq \gamma_1 \leq \ldots \leq \gamma_{T-1} \leq \gamma_T = 1$. Note that the 0-th intermediate posterior distribution ($\tilde{\pi}_0(\theta | Z)$) is simply the prior distribution $\pi(\theta)$ with
incorporation factor $\gamma_0 = 0$. Likewise, the $T$-th intermediate posterior distribution $\hat{\pi}_T(\theta | Z)$ is the full posterior distribution since $\gamma_T = 1$. At each time $t$, the target distribution is the $t$-th intermediate posterior distribution $\hat{\pi}_t(\theta | Z)$, and the prior is the intermediate posterior from the previous iteration $\hat{\pi}_{t-1}(\theta | Z)$.

At the end of each cycle, there still may be many replicates of a few unique particles, or sample impoverishment. To increase the number of unique particles, we “jitter” or “mutate” the particles through a carefully constructed kernel function (Gilks & Berzuini, 2001; Li et al., 2014; Liu & West, 2001). To increase the number of unique particles at the end of each cycle ($t$), we “jitter” or “mutate” the particles through a carefully constructed kernel function (Gilks & Berzuini, 2001; Li et al., 2014; Liu & West, 2001). Upon completion of the fast particle-based calibration algorithm, we are left with an ensemble of updated parameter sets (particles) which sensibly approximate the posterior distribution. Lee et al. (2020) also provides guidelines for choosing the number of cycles, how to mutate the particles, and how to construct these intermediate posterior distributions. We approximate the posterior distribution using “mutated” samples from the final ($T$-th) intermediate posterior distribution:

$$\hat{\pi}(\theta | Z) = \hat{\pi}_T(\theta | Z) = \sum_{i=1}^{N} w_T(\bar{\theta}^{(i)}) \delta(\bar{\theta}^{(i)})$$

(4)

where $\bar{\theta}^{(i)}$ is the $i$-th mutated particle, $w_T(\bar{\theta}^{(i)})$ are the corresponding weights from the $T$-th cycle, and $\delta(\bar{\theta}^{(i)})$ is a Dirac measure at $\bar{\theta}^{(i)}$.

3. Experimental Design

We demonstrate the approach for a case study in the Susquehanna River basin, Pennsylvania, United States. Pennsylvania provides a relevant study area as it ranked second, tenth, and fourteenth in the United States in terms of the frequency of flash flood-related fatalities, injuries, and casualties in 1959-2005 (Ashley & Ashley, 2008). This region has experienced several devastating flooding events over the recent decades, including floods associated with the remnants of Hurricane Ivan (September 2004), late winter–early spring extratropical systems (April 2005), warm-season convective systems (June 2006), and tropical storm Lee (September 2011) (Gitro et al., 2014; Grumm, 2011). In Pennsylvania, the Federal Emergency Management Agency (FEMA) paid $953 million in property damages to National Flood Insurance Program participants between 1975 and 2019 (FEMA, 2019).
We use the National Oceanic and Atmospheric Administration's (NOAA) Hydrology Laboratory-Research Distributed Hydrologic Model (HL-RDHM) (Koren et al., 2004). Distributed hydrologic modeling accounts for the spatial variability of model inputs, parameters and states to analyze rainfall-runoff processes at desired locations within a river basin. Distributed modelling involves processing and storing large amounts of data required to solve numerous and complex physics-based equations at each grid cell. We run HL-RDHM in a fully distributed mode at a spatial resolution of 2 km. The $2 \times 2$ km$^2$ resolution mainly allows for a more realistic representation of the stream network. Within HL-RDHM, we use the Sacramento Soil Moisture Accounting model with Heat Transfer (SAC-HT) (Koren et al., 2004) to represent hillslope rainfall-runoff processes, and the SNOW-17 module (Anderson et al., 2006) to represent snow accumulation and melt. SAC-HT is a physics-based, conceptual model where the basin system is divided into regularly spaced, square grid cells to account for spatial heterogeneity and variability. Each grid cell, in turn, is composed of storage components that store and transmit water. The cells are ultimately connected to each other through the stream network system, that is, each cell acts as a hillslope capable of generating surface and subsurface runoff that discharges directly into the streams. The hillslope runoff, generated at each grid cell by the SAC-HT and SNOW-17, is routed to the stream network using a nonlinear kinematic wave algorithm (Koren et al., 2004). Further information about the HL-RDHM model can be found for example in Koren et al. (2004), Reed et al. (2004), and Anderson et al. (2006). The HL-RDHM distributed hydrological model takes approximately 15 minutes per run on a single 2.3-GHz Intel Xeon E5-2697V4 (Broadwell) processor on the Cheyenne cluster (Computational and Information Systems Laboratory, 2017).

We use three main datasets: multisensor precipitation estimates, gridded near-surface air temperature, and streamflow. We use NOAA’s multisensor precipitation estimates and gridded near-surface air temperature products to run the hydrological model for parameter calibration purposes and to initialize the model. Multisensor precipitation estimates represent a continuous time series of hourly, gridded precipitation observations at $4 \times 4$ km$^2$ cells, which are produced by combining multiple radar estimates and in situ rain-gauge measurements (Prat & Nelson, 2015; Rafieeinasab et al., 2015). The gridded near-surface air temperature data are derived by combining multiple temperature observation networks, including the meteorological terminal aviation routine weather report (METAR), USGS stations, and National Weather Service Cooperative Observer Program (Siddique & Mejia, 2017). We use streamflow observations from the United States
We calibrate the model for the period of 2004-2008 and use 2009-2012 observations to evaluate the calibration performance. We use the year 2003 to spin up the model. As part of the calibration process, we select 12 out of the 17 model parameters associated with each model grid cell (Table S1). To improve calibration efficiency, basin-scale parameter multipliers, rather than the parameters in each grid, were calibrated and applied to the a-priori parameter grids (NWS, 2011). We only consider the model parameters that have a strong influence on the model output (see Figure S1). Exploring a higher-dimensional parameter space demands additional processors (particles) (Bain & Crisan, 2008; Jeremiah et al., 2011; Kantas et al., 2014) to sensibly calibrate the hydrological model. Selecting only the strongly influential model parameters can help reduce the computational costs considerably. This is, of course, an approximation and points to future research needs. The sensitive parameters are associated with different hydrodynamic processes related to baseflow, percolation, evaporation, snowfall, storm runoff, and channel routing (Table S1). These parameters are also suggested by several other studies (Gomez et al., 2019; Sharma et al., 2021; Siddique & Mejia, 2017; Zarzar et al., 2018) as the most sensitive parameters in the Susquehanna river basin.

We compare Bayesian calibration with relatively simple and low-cost model calibration approaches: i) stepwise line search (Kuzmin et al., 2008) and ii) precalibration (Edwards et al., 2011). Stepwise line search typically adjusts a subset of model parameters to minimize an objective function (e.g., root mean square error) and returns a single estimate of the model parameters (for details of the implementation please see Text S2) (Bowman et al., 2017; Carlberg et al., 2020; Fares et al., 2014; Mejia & Reed, 2011; Siddique & Mejia, 2017). Precalibration applies a screening criterion to a large ensemble of hydrologic model runs and rules out any implausible model runs that deviate substantially from the observations (refer Text S3 for the details) (Craig et al., 1997; Edwards et al., 2011; Holden et al., 2010; Williamson et al., 2017; Edwards et al., 2019). The Bayesian analogue to the stepwise line search could be the maximum \textit{a posteriori} estimate. However, both estimators use a different objective function and possibly result in different estimates. Precalibration is a sampling-based method without any specification of a likelihood function. These simple approaches of carrying out limited calibration are used by many academic studies (Rafieeinasab et al., 2015; Siddique and Mejia, 2017; Fares et al., 2014; Kim et
a., 2021; Mayr et al., 2013; Tarawneh et al., 2016; Li et al., 2019) as well as commonly practiced in real-word applications (Salas et al., 2018). They are used in part because they are simple and fast (Knutti et al., 2002; Reed et al., 2022).

We evaluate the calibrated model performance using several decision-relevant metrics. We use traditional deterministic metrics such as the Kling-Gupta Efficiency (KGE) (Mizukami et al., 2019), which provides a direct assessment of streamflow time series (e.g., shape, timing, water balance and variability) using the ensemble mean estimate. We also evaluate the probabilistic prediction skill using the Brier Skill Score (BSS) (Murphy, 1973) and the Continuous Ranked Probability Skill Score (CRPSS) (Murphy, 1970). The Brier score is essentially the mean squared error of the probability predictions, considering that the observation is one if the event occurs, and that the observation is zero if the event does not occur. The Continuous Ranked Probability Score measures the integral square difference between the cumulative distribution functions of the observation and predictions, averaged over all pairs of predictions and observations. The selection of these decision-relevant metrics is motivated by the balance between model output goodness-of-fit, calibration approaches, and data availability. The description of evaluation metrics is provided in Text S4 in the supporting information. Model calibration and evaluation is focused on high flows by choosing the river flow that exceeds NOAA’s Action Stage (McEnery et al., 2005). Action Stage refers to the stage which, when reached by a rising river, represents the level where the National Weather Service or a partner/user needs to take some mitigation action in preparation for possible significant hydrologic activity.

We assess the impact of model calibration on flood damage estimates. Flood damage represents interactions among hazard, exposure and vulnerability (Tellman et al., 2021; Wing et al., 2018). Hazard in this case refers to the magnitude of the flood event. Exposure characterizes property value in the floodplain. Vulnerability characterizes how sensitive the impacts are for a given hazard and exposure. We consider 2,000 hypothetical houses to quantify the damage from flood hazards (Figure S4; TextS6). We assess damage for a certain depth of water in a house by using a relatively simple Bathtub-based flood inundation model (Didier et al., 2019; Fereshtehpour & Karamouz, 2018; Neumann & Ahrendt, 2013; Yunus et al., 2016) and a vulnerability model (Scawthorn et al., 2006). The Bathtub model relies on a digital elevation model to provide flood depth in a house for a particular corresponding water level in the river (refer TextS5 and TextS6
for the details). We use a common vulnerability model (depth-damage function) provided by the Federal Emergency Management Agency (FEMA) (Scawthorn et al., 2006).

4. Results and Discussion

We first generate streamflow simulations using the "best" parameter estimates obtained via the stepwise line search (Figure 2). In the considered example, stepwise line search substantially underestimates the high streamflow (Figure 2). Stepwise line is designed to sample high-probability outcomes and excludes comprehensive sampling of the parametric distribution (Kuzmin et al., 2008; Sharma et al., 2019).

We account for parametric uncertainty using precalibration and FaMoS (Figure S1). Characterizing parametric uncertainty requires knowledge of model behavior throughout the (often high-dimensional) parameter space. Precalibration provides a relatively simple method to explore the high-dimensional parameter space. Precalibration is a low-cost way of ruling out implausible model runs. We begin with an initial ensemble of 5,000 model runs with input parameters settings selected from a 12-dimensional Latin hypercube design (Helton & Davis, 2003). We select an ensemble of 165 runs that fall within the +/- 75% window surrounding each observation. Note that specifying bounds for precalibration is a subjective choice (Craig et al., 1997; Edwards et al., 2011; Holden et al., 2010; Tarawneh et al., 2016). This choice impacts the “surviving” parameter samples. For instance, imposing tight bounds on the observed streamflow could lead to high-resolution sampling of the plausible parameter space and wider bounds may include more plausible runs into the final ensemble. We choose the considered acceptable range to sample into the upper tails of projected flood hazards, which are often associated with high-cost events.

FaMoS adopts a more complex (but also more powerful) calibration approach compared to precalibration. We incorporate domain-area expertise (prior distribution) of the unknown parameters and also account for additional sources of uncertainty such as model-observation discrepancies and observational error (see the Appendix for the details). As a result, we obtain a distribution of viable parameter values (posterior distribution) along with interval estimates, as opposed to a single best fit estimate (Figure S1). Unlike precalibration, FaMoS does not fix an arbitrary screening criterion, but rather uses a flexible statistical model to assess model-fit. Moreover, FaMoS sequentially explores the entire parameter space and systematically attempts to
move to a “target” region that contains the most plausible sets of model parameters. In contrast, precalibration attempts to locate this “target” region using a single initial ensemble of model runs.

Accounting for parametric uncertainty improves model performance metrics for the calibration data and out-of-sample predictions (Figure 3). We compute the skill score (KGE, BSS, and CRPSS) with reference to raw (uncalibrated) model runs using default parameter estimates obtained from several previous studies (Anderson et al., 2006; Reed et al., 2007). In terms of the performance metrics, model predictions remain skillful for all the calibration approaches (Figure 3). Precalibration outperforms the stepwise line search (best estimate predictions). Stepwise line search and precalibration are not designed to find the global optimum, which can lead to worse skill score as compared to FaMoS. FaMoS demonstrate a higher skill score than both the stepwise line search and precalibration for both calibration and out-of-sample evaluations.

Accounting for parametric uncertainty improves flood hazard estimates (Figure 4). The resulting predictive distribution of flood events demonstrates the impacts of model calibration. The stepwise line search underestimates the flood peaks by as much as 35% (Figure 4b) during calibration and 40% during out-of-sample prediction (Figure 4e). Precalibration captures the specific flood events, but exhibits very high prediction uncertainty as evidenced by the wider prediction intervals. Overall, FaMoS improves flood peak estimates and provides narrower prediction intervals. Consider, as an example, the case of Tropical Storm Lee with streamflow observation of 11, 292 m$^3$/sec (Figure 4). Precalibration provides a flood peak prediction of 10, 539 m$^3$/sec and prediction intervals (5%-95% credible interval) range from 6, 359 m$^3$/sec to 14, 222 m$^3$/sec (width = 7, 863 m$^3$/sec). FaMoS has a corresponding flood peak prediction of 11, 467 m$^3$/sec with a credible interval ranging from 9, 925 m$^3$/sec to 13, 121 m$^3$/sec (width = 3, 196 m$^3$/sec). Sampling both the parametric uncertainty and observation errors can improve the uncertainty characterization of model parameters and flood peak predictions (refer Supporting Information Figures S4 and S5). However, the prediction intervals can be quite wide, leading to arguably unrealistic streamflow estimates. Incorporating a flexible observational-error model and a discrepancy term may improve the representation of the underlying error structure.

We assess each calibration approach’s classification ability or how well each method discriminates between occurrences (water level crossing the action stage) versus non-occurrences (regular water level) of an event (Figure 5). Managing flood risks can require decision makers to choose between two options (e.g., to evacuate or not or to elevate a house or not) based on a
prediction of an event (e.g., water rising to a certain level) with one decision preferred if the event
doesn't occur, and the other if it does. A perfect prediction system for a binary outcome correctly
predicts the occurrence of an event (unity probability of detection) and never issues incorrect
predictions when it does not occur (zero probability of false detection). How well a prediction
system approaches this ideal case can be quantified by the relative operating characteristics (ROC)
curve (see Text S4) (Mason & Graham, 2002). Technically, the ROC curve assesses the quality of
probability predictions by relating the probability of detection (true alarm) to the corresponding
probability of false detection (false-alarm rate), as a decision threshold is varied across the full
range of a continuous prediction quantity (Figure 5). We fix the threshold corresponding to the
river flow that exceeds NOAA’s Action Stage. Streamflow predictions obtained using FaMoS
parameter distribution exhibit better discriminatory ability (higher ROC score) than the stepwise
line search and precalibration. Stepwise line search shows a relatively poor ability to discriminate
between different events. This poor ability to discriminate between the events can lead to poor
decisions and outcomes.

Neglecting parametric uncertainty also underestimates potential flood damage (Figure 6). We
find that the stepwise line search tends to underestimate the flood damage. The underestimation
bias increases as flood magnitude increases. Accounting for parametric uncertainty improves the
damage estimates for the calibration data and out-of-sample predictions. The damage credible
interval obtained using FaMoS parameter distribution generally captures the observed damage for
different flood events. As expected, at the upper tails of the damage, the predictive uncertainty
tends to be higher for the out-of-sample prediction as compared to the calibration.

5. Caveats

We use a relatively simple model and small region with hypothetical exposure to demonstrate
our points. This parsimony helps with transparency, but it comes with several caveats. For
example, our analysis focuses on calibrating high flows by choosing the river flow that exceeds
NOAA’s Action Stage. Temporal independence, conditioned on the model outputs, is a key
assumption within the calibration framework. We calibrate multiple disjoint (or unconnected)
instances of extreme streamflow events. We compute skill score to assess the performance of
different calibration approaches. However, implementing the Ljung-Box test and other diagnostic
tools for autocorrelation (Smith et al., 2015) would require calibrating a continuous streamflow
Future work might consider calibrating a continuous time series of streamflow, including low flows and moderate flows. Due to a large number of low and moderate flow observations, dimension-reduction techniques like principal components (Chang et al., 2014; Higdon et al., 2008) or eigenfunctions (Mak et al., 2018) may be appropriate to summarize the large datasets. This study samples shallow uncertainty about hydrologic model parameters as a case-study. There are, of course, other deep uncertainties (Lempert, 2002) affecting flood hazards and risks that could be taken into account in future work (Mendoza et al., 2015, Bates et al., 2021, Reed et al., 2022). These include model structural uncertainty, different spatial resolutions, land surface characteristics, or projections of the socio-economic systems (Gupta et al., 2012; Kavetski et al., 2006; Zarekarizi et al., 2020). Characterizing the individual uncertainty sources and their propagation is crucial to improve the reliability of flood hazard and risk projections. Increasing the spatio-temporal resolutions may drastically raise the hydrologic model’s complexity as well as the associated single model run times. To reduce the number of sequential hydrologic model evaluations, we can embed parallel Markov Chain Monte Carlo approaches such as Multiple-Try Metropolis (Liu et al., 2000) or “emcee” samplers (Goodman & Weare, 2010) or genetic algorithms (Park et al., 2009) into FaMoS calibration framework. We note that our damage estimates are based on a simple Bathtub-based flood inundation model. Future work could use process-informed models to characterize the impacts of hydrodynamic processes in damage estimates (Brunner, 1995; Coulthard et al., 2013; Judi et al., 2018). In addition, future work could sample the uncertainty surrounding the flood vulnerability of the building (Wing et al., 2020).

Although the objective of this study is not to compare different complex calibration approaches, FaMoS can add to emerging research into uncertainty quantification of a distributed hydrologic model. We demonstrate the ability of FaMoS to calibrate a large number of extreme flood events and consider a relatively larger river basin than in the several previous studies (Vrugt et al., 2008; Vrugt et al., 2009; Laloy and Vrugt, 2012). Computationally, the problem becomes very different to run and calibrate a spatially distributed model over a large river basin. Future study could compare FaMoS with other complex and state-of-the art Bayesian calibration approaches (e.g., Vrugt et al., 2008).

6. Conclusions
We use a Bayesian data-model fusion framework to calibrate a distributed hydrologic model and to demonstrate practical implications of neglecting key uncertainties on hazard- and risk-estimates. We compare the results of the Bayesian approach to two simpler methods: stepwise line search and precalibration. We show that these simpler methods can considerably underestimate flood hazards and risks. Precalibration improves flood hazards estimates over the best fit estimates, but provides a wider predictive interval (i.e., highly uncertain estimates) than the Bayesian approach. The predictive skill of the Bayesian approach dominates the stepwise line search and precalibration approaches. We show how neglecting model parametric uncertainty can substantially underestimate flood hazards and risk estimates and demonstrate how applying state-of-the-art statistical methods can help to refine flood-risk projections.

Acknowledgments

This study was co-supported by the US Department of Energy, Office of Science through the Program on Coupled Human and Earth Systems (PCHES) under DOE Cooperative Agreement No. DE-SC0016162 and DE-SC0022141 as well as the Penn State Center for Climate Risk Management. We thank Rob Nicholas, Skip Wishbone, Dave Judi, and the PSIRC team for inputs. All errors and opinions (unless cited) are those of the authors and not of the funding entities.

Disclaimer and License

The results, data, software tools, and other resources related to this work will be available under the GNU general public open-source license, as-is, without warranty of any kind, expressed or implied. In no event shall the authors or copyright holders be liable for any claim, damages, or other liability in connection with the use of these resources. This is academic research and not designed to be used to guide a specific decision.

Author contributions

All authors contributed to the study design. S.S. led the hydrologic analysis. B.L. and M.H. constructed the particle-based calibration model. I.H.S led the flood damage analysis. I.H.S. performed a code review. S.S., B.L, and K.K wrote the initial draft of the manuscript. All authors revised and edited the manuscript.
Data and Code Availability
The code used for this analysis and the data required to plot the results is available through a publicly accessible GitHub repository and under the GNU open-access license upon acceptance to a peer-reviewed journal. Reviewers can access these resources from https://github.com/beneec55/FamosHydroModel. All data and code currently available at GitHub will be published via Zenodo upon article acceptance.

Competing interests
The authors are not aware of any competing financial or nonfinancial interests.

Materials & Correspondence
Correspondence and requests for materials should be addressed to the corresponding author.
The code and data are available on GitHub (made public upon acceptance of the paper).
Figure 1: Diagrammatic representation of distributed hydrological model calibration framework. The framework also demonstrates flood hazards and risk components.
Figure 2: Historical time series of water level observation and model simulations obtained using best parameter estimates (stepwise line search). We obtain the observation from the United States Geological Survey (USGS) gauge records for ID 01554000 located upstream of Selinsgrove, Pennsylvania, USA. The most destructive floods in the Susquehanna River basin that occurred in recent years, each associated with different flood-generating mechanisms, includes Hurricane Ivan (September 2004), late winter–early spring extratropical systems (April 2005), warm-season convective systems (June 2006), and tropical storm Lee (September 2011).
Figure 3: Performance metrics for hydrological model calibration and out-of-sample prediction. We compute Kling-Gupta Efficiency (KGE), and Brier skill score (BSS), and mean Continuous ranked probability skill score (CRPSS). All the metrics are computed with reference to the default parameter set available from several previous studies (Anderson et al. 2006, Reed et al. 2004). Any positive values of the skill score, from 0 to 1, indicate that the calibration approach performs better than the reference system. Thus, a skill score of zero indicates no skill, and a skill of one indicates perfect skill. We plot the average value to compute KGE. CRPSS measures the integrated squared difference between the cumulative distribution function (cdf) of a model prediction, and the corresponding cdf of the observations. The CRPSS is averaged across n pairs of model predictions and observations, which leads to the mean CRPSS. BSS measures the averaged squared error of a probability prediction.
Figure 4: (a) - (c) Calibration and (d) - (f) and out-of-sample prediction for different flood events.
Figure 5: Relative operating characteristics (ROC) curve for different calibration approaches. ROC curve plots the probability of detection against the probability of false detection for a range of forecast probability levels. The ROC measures the trade-off between the fraction of ensemble that correctly predict the occurrence of an event (probability of detection) and the fraction that incorrectly predict its occurrence (probability of false detection). A larger area under the ROC curve represents a more skillful prediction, with more ability to discriminate between flood thresholds. The area under the ROC curve can range between 0 and 1, where a score of 1 implies perfect discrimination and a score of 0.5 or less implies predictive discrimination that is no better than a random guess. We also compute the ROC score. The ROC score measures the average gain over climatology for all probability levels. The ROC score for stepwise line search, precalibration and FAMOS is 0.55, 0.85 and 0.96 respectively.
Figure 6: Survival function (one minus the cumulative frequency) for damage estimates using streamflow obtained using the best parameter set (stepwise line search) and parameter distribution (FaMoS). We show damage estimates for a) calibration and b) out-of-sample prediction. cdf = cumulative distribution function.
Appendix A: Fast Model Calibrations (FaMoS) Details

1 Bayesian Calibration Framework

Suppose we have an observed time series $Z = (Z(r_1), ..., Z(r_n))^\prime$ times $r_i \in \mathcal{R}$ where $\mathcal{R}$ is the temporal domain of the process. We also have a deterministic computer model that generates a temporal process, or time series, at times $r_i \in \mathcal{R}$. Let $Y(r, \theta)$ be the computer model output at the time $r \in \mathcal{R}$ and the parameter (input) setting $\theta \in \Theta \subseteq \mathbb{R}^d$. $\Theta$ is the parameter space of the computer model with integer $d$ being the number of input parameters. In this study, we use a discontinuous temporal domain at $R$ distinct time points $\nabla = (r_1, ..., r_R)^\prime$. The vector $Y(\theta_i) = (Y(r_1, \theta_i), ..., Y(r_R, \theta_i))^\prime$ is the computer model output corresponding to parameter setting $\theta_i$. For input parameter setting $\theta$, we model the observations $Z$ as:

$$Z = Y(\theta) + \delta + \epsilon,$$

where $\epsilon \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$ are the independently and identically distributed observational error, and $\delta \in \mathbb{R}^n$ is a systemic data-model discrepancy term, which can be modeled as a zero-mean Gaussian process (Bhat et al., 2010; Bayarri et al., 2007) or other flexible functional forms (Brynjarsdottir and O’Hagan, 2014).

In the Bayesian calibration framework, we obtain samples (via a Markov chain Monte Carlo (MCMC) algorithm) from the posterior distribution:

$$\tilde{\pi}(\theta, \sigma^2, \delta | Z) \propto L(\theta, \sigma^2, \delta | Z)\pi(\theta)\pi(\sigma^2)\pi(\delta),$$

where $L(\theta, \sigma^2, \delta | Z)$ denotes the likelihood function and $\pi(\cdot)$ represents the prior distribution for the respective parameters and discrepancy term. Note that each evaluation of $L(\theta, \sigma^2, \delta | Z)$ requires running the computer model using specific input parameters $\theta$. Hence, MCMC-based calibration approaches are sensible for computer models with shorter single model run walltimes, typically under 5 seconds per model run (Lee et al., 2020). For our study, we estimate that a standard MCMC-based calibration approach would on the order of years to approximate the posterior distribution $\tilde{\pi}(\theta, \sigma^2, \delta | Z)$. We estimated the time using the Metropolis-Hastings algorithm with an all-at-once update for the model parameters ($\theta$). The dominating cost lies in evaluating the likelihood function, which requires running the hydrological model. For our case study, the HL-RDHM distributed hydrological model takes approximately 14.7 minutes per run on a single 2.3-GHz Intel Xeon E5-2697V4 (Broadwell) processor. Based on these values, 100,000 iterations of the all-at-once Metropolis-Hastings algorithm at 14.7 minutes per iteration would take 2.79 years of wall-time.

2 Particle-based Calibration Framework

We calibrate the HL-RDHM distributed hydrological model using the fast particle-based approach from Lee et al. (2020), which is built upon traditional Sequential Monte Carlo algorithms (Del Moral et al., 2006; Doucet et al., 2000; Liu and West, 2001), notably the Iterated Batch Importance Sampling (IBIS) (Chopin, 2002; Crisan and Doucet, 2000) method.
This method approximates the posterior distribution $\tilde{\pi}(\theta, \sigma^2, \delta|Z)$ using an evolving ensemble of particles.

We simplify the notation for an arbitrary target distribution as $\pi(\theta)$ with random variable $\theta \in \mathbb{R}^d$. In the hydrological model calibration framework, the target distribution $\pi(\theta)$ would be the posterior distribution $\tilde{\pi}(\theta, \sigma^2, \delta|Z)$ with random variables $\theta, \sigma^2$, and $\delta$ and observations $Z$. Suppose we want to estimate $\mu = E_\pi(g(\theta))$. Given $q(\theta) > 0$ whenever $g(\theta)\pi(\theta) > 0$, $\forall \theta \in \Theta$. Then $E_n[g(\theta)] = E_q[g(\theta)w(\theta)]$, where $w(\theta) = \frac{\pi(\theta)}{q(\theta)}$ is the importance weight and $\sum_{i=1}^N w(\theta_i) = 1$. The importance sampling estimator is $\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^N g(\theta_i)w(\theta_i)$ and $\hat{\mu}_n \rightarrow \mu$ with probability 1 as $n \rightarrow \infty$ by the strong law of large numbers. For target distributions with an unknown normalizing constant, the weights can be normalized as follows:

$$\tilde{w}(\theta_i) = \frac{w(\theta_i)}{\sum_{j=1}^n w(\theta_j)} = \frac{\pi(\theta_i)/q(\theta_i)}{\sum_{j=1}^n w(\theta_j)} \quad (A3)$$

where $\sum_{i=1}^N \tilde{w}(\theta_i) = 1$.

Sampling-Importance-Resampling (Gordon et al., 1993; Doucet et al., 2001) approximates a target distribution $\pi(\theta)$ with an empirical distribution of the particles $\hat{\pi}(\theta)$ from an importance function $q(\theta)$ such as the prior distribution. The empirical distribution $\hat{\pi}(\theta)$ is defined as:

$$\hat{\pi}(\theta) = \sum_{i=1}^N \tilde{w}(\theta_i)\delta(\theta_i) \approx \pi(\theta), \quad (A4)$$

where $\tilde{w}(\theta_i)$ are the normalized importance weights, $\delta(\theta_i)$ is a Dirac measure that places unit mass at $\theta_i$ and $\sum_{i=1}^N \tilde{w}(\theta_i) = 1$.

Poor choices of importance functions can lead to inaccurate approximations of the target distribution (Doucet et al., 2000) where the bulk of the particles $\theta_i$’s do not reside in the high-probability regions of the target distribution $\pi(\theta)$. Weight degeneracy occurs when the vast majority of the particles have near-zero importance weights. Multinomial resampling methods can combat weight degeneracy by eliminating the particles with very small important weights and replicating those with higher weights (Gordon et al., 1993; Doucet et al., 2000). After resampling, we reset all importance weights such that $w(\theta_i) = 1/N$ and use the unweighted empirical distribution $\tilde{\pi}(\theta)$:

$$\tilde{\pi}(\theta) = \frac{1}{N} \sum_{i=1}^N N_i \delta(\theta_i), \quad (A5)$$

where $N_i$ is the number of replicates corresponding to particle $\theta_i$ and $\sum_{i=1}^N N_i = N$. Extreme weight degeneracy, where very few particles have any significant weight, can lead to sample impoverishment where a few unique particles $\theta_i$’s are heavily replicated in the re-sampling step; hence, the empirical distribution $\tilde{\pi}(\theta)$ may poorly approximate the target distribution $\pi(\theta)$.

An alternative method mutates the replicated particles with samples from $K(\theta_i^{(t-1)})$, the Metropolis-Hastings transition kernel (Gilks and Berzuini, 2001), whose stationary distribution is also the target distribution $\pi(\theta)$. The mutation stage proceeds with $K$ Metropolis-Hastings updates for each particle $\theta_i$, for $i = 1, ..., N$. Alternative mutation schemes use
genetic algorithms (Zhu et al., 2018) or different families of transition kernels, $K(\cdot)$ (Papaioannou et al., 2016; Murray et al., 2016). We set the $K$-th sample drawn via MCMC as the mutated particle $\tilde{\theta}_i$. Since $\tilde{\theta}_i \sim \pi(\theta)$, the resulting empirical distribution $\hat{\pi}(\theta)$ approximates the target distribution $\pi(\theta)$:

$$\pi(\theta) \approx \hat{\pi}(\theta) = \sum_{i=1}^{N} \delta(\tilde{\theta}_i). \quad (A6)$$

Unfortunately, poor importance functions can result in severe sample impoverishment, which may require very long (and costly) mutation stages to provide an accurate representation of the target distribution (Li et al., 2014). Mixture approximations (Gordon et al., 1993) or kernel smoothing methods (Liu and West, 2001) can mutate or rejuvenate the replicated particles. However, these methods may not scale well to high-dimensional target distributions (Doucet et al., 2000).

2.1 Fast Particle-based Approach For Computer Model Calibration

In this study, we aim to approximate the posterior $\bar{\pi}(\theta, \delta, \sigma^2_t|Z)$ from a computationally efficient approach. The fast particle-based approach (Lee et al., 2020) utilizes a set of tempered, or intermediate, posterior distributions $\bar{\pi}_t(\theta, \delta, \sigma^2_t|Z)$ for $t = 1, ..., T$, which will act as both the importance functions and target distributions. Intermediate posterior distributions can be generated using likelihood tempering (Chopin, 2002; Neal, 2001; Liang and Wong, 2001) where the $t$th intermediate posterior distribution is defined as:

$$\bar{\pi}_t(\theta, \delta, \sigma^2_t|Z) \propto L(\theta, \delta, \sigma^2_t|Z)^{\gamma_t} \pi(\theta) \pi(\delta) \pi(\sigma^2_t), \quad (A7)$$

where $\gamma_t$’s are determined according to a schedule where $\gamma_0 = 0 < \gamma_1 < \cdots < \gamma_T = 1$. For each $\bar{\pi}_t(\theta, \delta, \sigma^2_t|Z)$, the likelihood component is a fractional power of the original likelihood $L(\theta, \delta, \sigma^2_t|Z)$. Using an adaptive incorporation schedule (Lee et al., 2020), we can select the appropriate $\gamma = \{\gamma_0, \gamma_1, ..., \gamma_T\}$ within the calibration algorithm.

For cycle $t = 1$, we set the importance distribution to be the prior distribution $p(\theta, \delta, \sigma^2_t) = p(\theta)p(\delta)p(\sigma^2_t)$, and the target distribution to be the first intermediate posterior distribution, $\pi(\theta, \delta, \sigma^2_t|Z)$. For subsequent cycles $t$, the importance distribution is $\pi_{t-1}(\theta, \delta, \sigma^2_t|Z)$ and the target distribution is $\bar{\pi}_t(\theta, \delta, \sigma^2_t|Z)$.

Next, we mutate the particles via short runs of the Metropolis-Hastings algorithm, where the stationary distribution is $\pi_t(\theta, \delta, \sigma^2_t|Z)$, the $t$-th intermediate posterior distribution. Note that the importance and target distributions are consecutive ($t$-th and $(t+1)$-th) intermediate posterior distributions, so there is considerable overlap between the high-probability regions of the two distributions. In the mutation stage, we employ the stopping rule from Lee et al. (2020) to control the number of Metropolis-Hastings updates; thereby preventing any unnecessary computer model runs. The mutation stages ends when the Bhattacharyya distance (Bhattacharyya, 1946) between two sets of particles from the mutation stage stabilizes.
2.2 Adaptive incorporation schedule

To reduce computational costs and potentially reduce unnecessary computer model evaluations, we adopt the adaptive incorporation schedule from Lee et al. (2020). For avoid confusion, we simplify the notation in this subsection by defining $\theta = (\theta, \sigma^2, \delta)$, the combined vector of unknown parameters. Upon initialization, we set the first incorporation increment $\gamma_0 = 0$. We draw the initial set of particles $\hat{\theta}_0$ from $\pi_0(\theta | Z) \propto L(\theta | Z)^0 \pi(\theta) = \pi(\theta)$, the prior distribution of model parameters. For the subsequent cycles $t = 1, 2, 3, ...$, we calculate the full likelihood $L(\hat{\theta}_{t-1}^(i) | Z)$ for $i = 1, ..., N$ where $\hat{\theta}_{t-1}^(i)$ denotes the parameter samples from the previous cycle $t - 1$. Next, we compute the optimal $\gamma_t$ that returns an effective sample size (ESS) of $ESS_{\text{thresh}}$ or a sample size closest to $ESS_{\text{thresh}}$: $\gamma_t = \arg\min_\gamma \{(ESS_\gamma - ESS_{\text{thresh}})^2\}$, where $\gamma \in (\gamma_{\text{min}}, 1 - \gamma_{t-1})$, $\gamma_{\text{min}}$ is a previously set minimum incorporation value, $ESS_\gamma = \sum_{i=1}^N 1/w_t(\hat{\theta}_{t}^{(i)})^2$, and $w_t(\hat{\theta}_{t}^{(i)}) \propto L(\hat{\theta}_{t}^{(i)} | Z)^\gamma$. Note that we can lower computational costs by evaluating the full likelihood $L(\hat{\theta}_{t}^{(i)} | Z)$ only once before the optimization.

We stop the scheduling algorithm when $\sum_{i=1}^t \gamma_t = 1$, or when the entire likelihood has been incorporated and the target distribution evolves to the full posterior distribution $\pi(\theta | Z)$. Note at each cycle $t$, we set the incorporation increment ($\gamma_t$) to be between $\gamma_{\text{min}}$ and $1 - \sum_{i=1}^t \gamma_t$. The user will typically set the minimum incorporation increment $\gamma_{\text{min}}$ and the threshold effective sample size, $ESS_{\text{thresh}}$. We provide our choice of $\gamma_{\text{min}}$ and $ESS_{\text{thresh}}$ in the next section (Implementation Details).

Adaptive likelihood incorporation schedule

1. Initialization: At $t = 0$, set $\gamma_0 = 0$.
2. When $t > 0$ and $\sum_{i=1}^{t-1} \gamma_i < 1$
   - Compute $L(\hat{\theta}_{t-1}^{(i)} | Z)$ for $i = 1, ..., N$
   - Set $\gamma_t = \arg\min_\gamma \{(ESS_\gamma - ESS_{\text{thresh}})^2\}$, where $ESS_\gamma = \sum_{i=1}^N 1/w_t(\hat{\theta}_{t}^{(i)})^2$, $w_t(\hat{\theta}_{t}^{(i)}) \propto L(\hat{\theta}_{t}^{(i)} | Z)^\gamma$, and $\gamma \in (\gamma_{\text{min}}, 1 - \gamma_{t-1})$.
   - Update $t \leftarrow t + 1$
3. When $\sum_{i=1}^t \gamma_i = 1$: Stop Calibration

2.3 HL-RDHM Calibration: Implementation Details

We now return to the original notation of the unknown parameters $\theta, \sigma^2, \delta$. The target distribution is the full posterior distribution $\pi(\theta, \sigma^2, \delta | Z)$ and the Bayesian hierarchical framework for the HL-RDHM distributed hydrological model calibration is as follows:

Data Model: $Z | Y(\cdot), \theta, \delta, \sigma^2 \sim \mathcal{N}(Y(\theta) + \delta, \sigma^2 I)$  \hspace{1cm} (A8)
Process Model: $\delta | \sigma^2 \sim \mathcal{N}(0, \sigma^2 I)$  \hspace{1cm} (A9)
Parameter Model: $\theta \sim \pi(\theta), \sigma^2 \sim \pi(\sigma^2), \sigma^2 \sim \pi(\sigma^2)$  \hspace{1cm} (A10)
Algorithm 1: Fast Particle-based Calibration

Data: $Z$
Initialization: 
Draw $\tilde{\theta}_0^{(i)} \sim p(\theta)$ for particles $i = 1, \ldots, N$.
Set $w_0^{(i)} = 1/N$, $\gamma_0 = 0$, and $K$;

for cycles $t = 1, \ldots, T$ do

1. Compute full likelihood: 
   Calculate $L(\tilde{\theta}_t^{(i)}|Z)$ for $i = 1, \ldots, N$;

2. Select optimal likelihood incorporation increment $\gamma_t$: 
   Set $\gamma_t = \text{argmin}_\gamma \{(ESS\gamma_t - ESS_{\text{thresh}})^2\}$, where $\gamma \in (0.1, 1 - \sum_{h=1}^{t-1} \gamma_h)$
   Note: $ESS\gamma_t = \sum_{i=1}^{N} \frac{1}{w^{(i)}_t^2}$ and $w_t^{(i)} \propto L(\tilde{\theta}_t^{(i)}|Z)^\gamma_t$;

3. Compute importance weights: 
   $w_t^{(i)} \propto w_{t-1}^{(i)} \times L(\tilde{\theta}_t^{(i)}|Z)^{\gamma_t}$;

4. Re-sample particles via multinomial sampling: 
   Draw $\tilde{\theta}_t^{(i)}$ from $\{\tilde{\theta}_{t-1}^{(1)}, \ldots, \tilde{\theta}_{t-1}^{(N)}\}$ with probabilities $\propto \{w_t^{(1)}, \ldots, w_t^{(N)}\}$;

5. Set intermediate posterior distribution: 
   Set $\pi_t(\theta|Z) \propto L(\theta|Z)^{\tilde{\gamma}} \pi(\theta)$, where $\tilde{\gamma} = \sum_{j=1}^{t} \gamma_j$;

6. Mutation: 
   Using each particle $\tilde{\theta}_t^{(1)}, \ldots, \tilde{\theta}_t^{(N)}$ as the initial value, run $N$ chains of an MCMC algorithm with target distribution $\pi_t(\theta|Z)$ for $2K$ iterations

7. Check stopping criterion: 
   Compute $\delta_B = DB(h(\tilde{\theta}_t^{K}), h(\tilde{\theta}_t^{2K}))$;
   if $\delta_B < \varepsilon_B$ then 
     Set $\tilde{\theta}_t^{(i)} = \tilde{\theta}_t^{(i),2K}$;
   else 
     Run $K$ additional updates and re-evaluate stopping criterion
     Continue until stopping criterion is met

8. Stop when full likelihood is incorporated: 
   if $\sum_{i=1}^{N} \gamma_t = 1$ then 
     End Algorithm;
   else 
     Reset weights: $w_t^{(i)} = 1/N$ for particles $i = 1, \ldots, N$;
     Set $t=t+1$ and return to Step 1;
where $\pi(\theta), \pi(\sigma_\theta^2)$, and $\pi(\sigma_\epsilon^2)$ denote the prior distributions of $\theta$, $\sigma_\theta^2$, and $\sigma_\epsilon^2$, respectively. For $\pi(\theta)$, we place a priori independent uniform priors on each of the model parameters with ranges (lower and upper bounds) based on domain-area expertise.

Instead of estimating the nuisance parameters $\sigma_\theta^2$ and $\sigma_\epsilon^2$ separately, we chose to combine these as $\sigma^2 = \sigma_\theta^2 + \sigma_\epsilon^2$. We place a standard non-informative inverse gamma prior on the combined error variance $\sigma^2 \sim IG(0.2, 0.2)$. Note that we assume conditional independence among the extreme observations given the model outputs. The updated Bayesian hierarchical framework is:

\begin{align}
\text{Data Model:} \quad & Z|Y(\cdot), \theta, \sigma^2 \sim N(Y(\theta), \sigma^2 I) \quad \text{(A11)} \\
\text{Parameter Model:} \quad & \theta \sim \pi(\theta), \quad \sigma^2 \sim \pi(\sigma^2) \quad \text{(A12)}
\end{align}

While much of the fast particle-based approach is automated, the user must select the: (1) total number of particles, $N$; (2) baseline number of Metropolis-Hastings updates run before checking the stopping criterion, $K$; (3) minimum incorporation $\gamma_{min}$ at each cycle; and (4) the effective sample size threshold $ESS_{thresh}$. We chose $N = 2015$ particles based on the available resources. On the Cheyenne HPC, this requires 56 nodes with 36 processors per node. For the stopping criterion, we use $K = 7$ as the baseline length. The floor for the incorporation increment is fixed at $\gamma_{\min} = 0.1$ such that we incorporate at least $L(\theta|Z)^{0.1}$ into the intermediate posterior at each cycle. Finally, the $ESS_{thresh} = N/2$, which is the typical threshold that activates resampling in many sequential Monte Carlo methods (Del Moral et al., 2006). We calibrate the HL-RDHM distributed hydrological model using Cheyenne (Computational and Information Systems Laboratory, 2017), a 5.34-petaflops high performance computer operated by the National Center for Atmospheric Research (NCAR).

We employ message passing interface (MPI) and the R package Rmpi for any parallelized operations such as computing importance weights and particle mutation.

Consider the vector of HL-RDHM model parameters $\theta = (\theta_1, ..., \theta_12)'$. The prior distribution $\pi(\theta_j)$ for the $j$-th model parameters follow a univariate uniform distribution with lower and upper bounds specified by our hydrological model experts. $\theta_j \sim Unif(l_j, u_j)$ with hyperparameters $l_j$ (lower bound) and $u_j$ (upper bound) specified in Table S1. We place a standard non-informative inverse gamma prior on the combined error variance $\sigma^2 \sim IG(\alpha_{\sigma^2}, \beta_{\sigma^2})$ where $\alpha_{\sigma^2} = 0.2$ and $\beta_{\sigma^2} = 0.2$. 
References


References

https://doi.org/10.1002/2016ef000485


https://doi.org/10.1002/2015wr016967


Beven, K. J., and A. M. Binley, The future of distributed models: Model calibration and


https://doi.org/10.1142/9781848162716_0006


https://doi.org/10.1137/130916138


Gramacy, R. B. (2020). *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. Chapman and Hall/CRC.


81(5), 345–354.

https://doi.org/10.1029/2008wr006824


https://doi.org/10.1029/2010wr010217


Kantas, N., Beskos, A., & Jasra, A. (2014). Sequential Monte Carlo Methods for High-


Environments Prone to Channel Infiltration. *Journal of Hydrometeorology*, 22(11), 2939–2969.


Mason, S. J., & Graham, N. E. (2002). Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. Quarterly Journal of the Royal Meteorological Society, 128(584), 2145–2166.


Water Resources Research, 51(8), 6381–6398.


https://doi.org/10.1002/2017ef000607


