Bayesian calibration of a flood inundation model using spatial data

Jim W. Hall,[1] Lucy J. Manning,[2] and Robin K. S. Hankin[3]

Received 20 August 2009; revised 16 October 2010; accepted 20 December 2010; published 21 May 2011.

[1] Bayesian theory of model calibration provides a coherent framework for distinguishing and encoding multiple sources of uncertainty in probabilistic predictions of flooding. This paper demonstrates the use of a Bayesian approach to computer model calibration, where the calibration data are in the form of spatial observations of flood extent. The Bayesian procedure involves generating posterior distributions of the flood model calibration parameters and observation error, as well as a Gaussian model inadequacy function, which represents the discrepancy between the best model predictions and reality. The approach is first illustrated with a simple didactic example and is then applied to a flood model of a reach of the river Thames in the UK. A predictive spatial distribution of flooding is generated for a flood of given severity.

Citation: Hall, J. W., L. J. Manning, and R. K. S. Hankin (2011), Bayesian calibration of a flood inundation model using spatial data, *Water Resour. Res., 47*, W05529, doi:10.1029/2009WR008541.

1. Introduction

[2] The need to estimate and communicate uncertainty in predictions of flood extent and estimates of flood risk is now widely appreciated [*Krzysztofowicz*, 2001; *Todini*, 2004, 2007]. Decision-makers can legitimately expect technical specialists to provide and justify uncertainty estimates so that they can make risk-based decisions that account for uncertainty [*Hall and Solomatine*, 2008]. This paper deals with the ubiquitous problem of uncertainty analysis in the use of (usually quite complex) hydraulic computer models to predict flooding. Emphasis is upon the prediction of flood risk, as a basis for planning or design decisions, as opposed to flood forecasting, which is the more usual focus of papers on flood model uncertainty. The flood risk analysis problem [*Dawson et al.*, 2005; *Apel et al.*, 2008] brings with it the luxury of not having to do computations in real time, but the burden of having to estimate flood damages over the full distribution of possible hydrological boundary conditions and system states, which may include failed and non-failed states of complex dike systems or control structures [*Dawson and Hall*, 2006]. The motive of uncertainty analysis is even more cogent in the context of risk analysis than in the context of flood forecasting [*Hall*, 2003] because of the deliberative and often contested nature of strategic flood risk management decisions. In establishing strategies for flood risk management, decision-makers may wish to seek solutions that are robust to uncertainty and properly account for their attitudes toward risk [*Hall and Solomatine*, 2008].

[3] This paper seeks to address the same statistical objective of several previous papers in adopting a Bayesian treatment of the various sources of uncertainty in flood predictions, addressing separately observation errors, parameter uncertainties, and model structural uncertainties. The overall objective is to generate the probability that some scalar or vector observable quantity of interest ζ, such as the maximum water depth during a flood at some point in a river or floodplain, will be in some set $\mathcal{Z}$ i.e., $P(\zeta \in \mathcal{Z})$, where evidence upon which to base this predictive distribution includes field observations and predictions from computer models.

[4] In the hydrological community, considerable effort has been devoted to identification of separate error sources in the parameterization of conceptual rainfall-runoff models. *Kavetski et al.* [2006a, 2006b] and *Thyer et al.* [2009] have addressed the issue of input errors with a Bayesian calibration scheme, although *Renard et al.* [2010] pointed out that the joint estimation of input and structural errors gives rise to identifiability problems. Structural errors have been addressed primarily in the context of data assimilation [e.g., *Krzysztofowicz*, 1999; *Vrugt et al.*, 2005; *Moradkhani et al.*, 2005], or by combining outputs from different models [e.g., *Duan et al.*, 2007; *Todini*, 2008].

[5] A Bayesian formulation, addressing model structural errors, has been the subject of considerable recent research in the statistical literature [*Kennedy and O'Hagen,* 2001a, 2001b; *Goldstein and Rougier,* 2004, 2009; *Higdon et al.*, 2004; *Campbell*, 2006], which forms the basis for the following development. *Conti et al.* [2009] and *Liu and West* [2009] have sought to apply this work to the dynamic characteristics of rainfall-runoff models. In particular, this recent Bayesian thinking has tackled, from a statistical point of view, the use of physically based computer simulations to an extent that is far from traditional in the statistical literature. It seeks to construct a coherent account of how physically based computer models are informative about reality, and of the way in which it is possible to use observations and the expert intuitions of modelers and scientists to understand the inevitable discrepancies between model predictions and reality.

[6] To develop this Bayesian framework, consider a deterministic computer model $S(\upsilon, \theta)$, where υ is a vector

[1]Environmental Change Institute, University of Oxford, Oxford, UK.
[2]School of Civil Engineering and Geosciences, Newcastle University, Newcastle upon Tyne, UK.
[3]Department of Land Economy, University of Cambridge, Cambridge, UK.

0043-1397/11/2009WR008541

of boundary conditions, such as the upstream flow and channel geometry, as well as containing the coordinates x of points where predictions of flood depths are required. $S(\upsilon, \theta)$ is referred to as a "simulator" because it is seeking to reproduce the processes in reality influencing ζ. The response of $S(\upsilon, \theta)$, as well as being a function of boundary conditions υ, is also a function of some model parameter(s) θ, which may, for example, include friction parameterization of the river channel. While experienced modelers may have intuitions about approximate values of these parameters for a given river, they are not precisely measurable in nature, and so are often dealt with as calibration parameters. The parameters θ are taken as being constant (scalars or functionals) but unknown. The simulator is used to predict some quantity of interest (for example, flood depths) for different values of υ, where the vector υ contains inflow at the upstream boundary of the model and other relevant boundary conditions, as well as the coordinates of the points in the floodplain where prediction is required. Moreover, we may wish to predict flood depths under conditions in which υ has been changed in the model, for example, to represent proposed channel improvement or other engineering works.

[7] Because of deficiencies in the simulator, there is no value of θ for which $S(\upsilon, \theta)$ is a perfect representation of "reality" $\zeta(\upsilon)$. Instead, suppose that the model predictions are separated from reality by a "model inadequacy" function $\delta(\upsilon)$, which is the (unknown) discrepancy between the model and reality

$$\zeta(\upsilon) = S(\upsilon, \theta) + \delta(\upsilon). \tag{1}$$

[8] In Bayesian analysis all observable quantities are, in general, treated as being uncertain, and this approach is applied to equation (1), with the exception that the simulator S is, for the time being, treated as a deterministic function because it is a computer model that always returns precisely the same output for a given vector of inputs. Even this assumption will be relaxed in due course to address the situation where, because of limited computational resources, the simulator S can only be run at a limited number of points (υ, θ). Furthermore, it is assumed that the points υ can be specified precisely and without error. This seems to be reasonable in the context of prediction for given conditions defined by υ, though input errors will have to be incorporated in the context of calibration, and are dealt with toward the end of this paper.

[9] Uncertainty in prediction of $\zeta(\upsilon)$ (equation (1)) arises from the fact that θ and $\delta(\upsilon)$ are uncertain, represented by a joint probability distribution $f(\theta, \delta_\upsilon)$, in which case

$$P[\zeta(\upsilon) \in \mathcal{Z}] = \iint I_{\mathcal{Z}}[S(\upsilon, \theta) + \delta_\upsilon] f(\theta, \delta_\upsilon) d\theta d\delta_\upsilon, \tag{2}$$

where I_z is the indicator function of the set $\mathcal{Z}$.

[10] While traditionally "calibration" has meant a process of identifying point values for parameters θ and fixing those values for prediction, the deficiencies of this approach are now well known in the hydrological community [*Beven and Binley*, 1992; *Beven*, 2006]. The Bayesian version of calibration involves updating the prior distribution for the model parameters based on a comparison, via a likelihood function, of the simulator outputs with observation. In other words, calibration process involves using observations $z^\dagger$ at points $\upsilon^\dagger$ to update beliefs about θ and $\delta(\upsilon)$ to generate a revised estimate $P[\zeta(\upsilon) \in \mathcal{Z} | z^\dagger]$ of the quantity of interest. The observations will in general include observation error and, furthermore, it may be necessary to transform the data so that they are comparable with the simulator outputs (the commensurability problem [*Beven*, 2006]), which may be a further source of uncertainty. In order to represent output observation error, write

$$z^\dagger = \zeta(\upsilon^\dagger) + e^\dagger = S(\upsilon^\dagger, \theta) + \delta(\upsilon^\dagger) + e^\dagger, \tag{3}$$

where e is the (unknown) observation error, which is now added to the joint prior distribution $f(\theta, \delta_\upsilon, e)$ of uncertainty quantities, though e may in practice be taken to be independent of θ and δ_υ.

[11] $P[\zeta(\upsilon) \in \mathcal{Z} | z^\dagger]$ can be expanded in equation (2) so

$$P[\zeta(\upsilon) \in \mathcal{Z} | z^\dagger] = \iiint I_{\mathcal{Z}}[S(\upsilon, \theta) + \delta_\upsilon] f(\theta, \delta_\upsilon, e | z^\dagger) d\theta d\delta_\upsilon de \tag{4}$$

and from Bayes' rule

$$P[\zeta(\upsilon) \in \mathcal{Z} | z^\dagger] = c \iiint I_{\mathcal{Z}}[S(\upsilon, \theta) + \delta_\upsilon] f(z^\dagger | \theta, \delta_\upsilon, e) f(\theta, \delta_\upsilon, e) d\theta d\delta_\upsilon de, \tag{5}$$

where

$$c = \left[Pr(z = z^\dagger)\right]^{-1} = \left[\iiint f(z^\dagger | \theta, \delta_\upsilon, e) f(\theta, \delta_\upsilon, e) d\theta d\delta_\upsilon de\right]^{-1} \tag{6}$$

and $f(z^\dagger | \theta, \delta_\upsilon, e)$ is the likelihood function. Equation (5) forms the basis for calibrated prediction, which is the objective of this paper.

[12] Note that in equation (5) predictions are of reality, not of the distribution of future observations. Flood risk managers are interested in predicting true water levels rather than what their imperfect instruments measure those water levels to be. If the interest is in predicting observations (as for example, is done in the approach of *Montanari and Brath* [2004] or *Shrestha and Solomatine* [2006]), then the error term e should be added into the indicator function in equation (5).

[13] In order to proceed in practice with the Bayesian framework set out above, it is necessary to specify the prior distributions and likelihood functions and implement some practical method for computing the integrals. The model inadequacy function $\delta(\upsilon)$ has received rather limited attention in previous treatments of model uncertainty [*Thyer et al.*, 2009], yet represents the important and complex processes that separate best model predictions from reality. Specification of this function in statistical terms requires a careful compromise between flexibility (to do justice to the complexity of the processes it is supposed to represent) and identifiability based on often rather scarce observations.

Here, we make use of Gaussian processes as a reasonable compromise between complexity and parsimony, following previous statistical work in this area by *Kennedy and O'Hagan* [2001a] (referred to hereafter as KOH2001).

[14] In the following, first some technical details of the use of Gaussian processes are supplied. The Bayesian calibration approach is applied to a simple synthetic study to illustrate its principles and the practicalities of implementation. It is then applied to the calibration of a flood inundation model using synthetic aperture radar observations of a flood event. It is explained how input error can be incorporated within the Bayesian framework. The paper concludes with a discussion of the benefits and disadvantages of the proposed approach and suggestions for several areas of future development.

2. Theoretical Background

2.1. Gaussian Processes

[15] The joint distribution of a Gaussian process is an n-dimensional multivariate Normal. This representation can be extremely flexible, permitting the modeling of complex surfaces. Moreover, adoption of Gaussian processes as a framework for Bayesian modeling of uncertain functions means that use can be made of the well-known properties of the conditional Gaussian process (see for example, *Mardia et al.* [1979] or *Vanmarcke* [1983]). Here we work with the joint density function of the following hierarchical form $f(x) \sim \mathcal{N}[m(x), c(x,x')] : x \in \mathbb{R}^N$, where $m(x)$ is the mean function and $c(x, x')$ is the covariance function. A linear model is adopted for the mean function

$$m(x) = \sum_{j=1}^{p} \beta_j h_j(x), \tag{7}$$

where $h_1(x), \ldots, h_p(x)$ are p known basis functions and $\beta_1, ..., \beta_p$ are unknown coefficients to be estimated. The basis functions $h_j(x)$ may, for example, be low order polynomials. Experts may be expected to have prior beliefs about the form of these functions or at least some knowledge from which a prior might be constructed. Higher order functions may, in principle, be attractive but in practice will often be hard to identify. While a very simple basis function (such as $h(x) = 1$, which is used by KOH2001) may appear to be a rather unsubtle instrument with which to estimate the behavior of a complex computer model or indeed its discrepancy with reality, it will become clear that the posterior distribution of the Gaussian process is sufficiently flexible to reflect complex behavior remarkably accurately.

[16] KOH2001 adopt the commonplace covariance function for the Gaussian process of the form $c(x,x') = \sigma^2 r(x - x')$, where $r(\cdot)$ is a correlation function, with $r(0) = 1$. A common choice of correlation function, which ensures differentiability at all orders, is

$$c(x,x') = \sigma^2 \exp\left[-\sum_{i=1}^{N} \omega_i (x_i - x_i')^2\right] \tag{8}$$

with parameters $(\sigma^2,\ \omega_i)$. It should be noted that the greater the accuracy of the mean function, the less the effect of an inappropriate choice of covariance function.

2.2. Gaussian Process Emulators of Computer Codes

[17] The simulator $S(\upsilon, \theta)$ is often computationally expensive so only a few tens or hundreds of model realizations are available, whereas practical evaluation of the integrals in equations (4) to (6) may require many thousands of function evaluations. KOH2001 treats the simulator $S(\upsilon, \theta)$ as a deterministic but unknown function which is sampled in a series of computer experiments. The results from these experiments can be used to construct an emulator [*Oakley and O'Hagan*, 2002], which is a very fast statistical approximation to $S(\upsilon, \theta)$ that can be used to estimate the simulator output at points where it has not actually been run. There are a number of potential forms of the emulator function. For example, given sufficient model realizations it might be possible to train an artificial neural network or some other machine-learning algorithm to emulate the simulator output. KOH2001 employ a Gaussian process model as the emulator function, which has the desirable property that if the posterior distribution of the Gaussian process is used to estimate the simulator output at points where the simulator has been run, it will reproduce the computer model output exactly. In other words, the emulator will recover the training data without error, reflecting the fact that the computer code is taken to be a deterministic function. Away from the training points, the emulator generates mean and variance estimates of the unknown computer model outputs based on an assumption of smoothly varying output (equation (8)). In other words, subject to a correlation assumption, the emulator not only predicts model output at points where the computer model has not been run but also generates associated uncertainties, which are a function of the distance of the prediction point from the training points.

[18] The known points in the simulator's input space, at which the emulator is trained, are written as (υ, t) in order to distinguish from subsequent calibration and prediction when the calibration parameters θ are taken to be uncertain. The simulator $S(\upsilon, t)$ is approximated by an emulator $\eta(\upsilon, t) \sim \mathcal{N}\{m_\eta(\upsilon, t),\ c_\eta[(\upsilon, t), (\upsilon',\ t')]\}$, where the mean and covariance functions are in the same form as equations (7) and (8), respectively.

[19] *Oakley and O'Hagan* [2002] demonstrate that the emulator predictions have a conditional multivariate t distribution with respect to the posterior local mean $m^*(x)$ and covariance $c^*(x, x')$ and estimated $\hat{\sigma}$

$$\hat{\sigma}^{-1}\left(c^*(x,\ x')\frac{q-p-2}{q-p}\right)^{-0.5}[\eta(x) - m^*(x)] \sim t_{q-p}, \tag{9}$$

where $x = (\upsilon, t)$, q is the number of simulator outputs used to train the emulator, and t_{q-p} is the t distribution with $q-p$ degrees of freedom.

[20] To illustrate the behavior of a Gaussian process emulator, suppose that the computer code of interest implements the function $y = x\sin 8x$, but for illustrative purposes this function is treated as being "unknown." Figure 1 illustrates the emulator prediction having been fitted to 5, 7, 9, and 11 points sampled incrementally from this unknown function. It is clear how the emulator predicts without error at the training points and how the prediction interval narrows as the density of sample points increases.

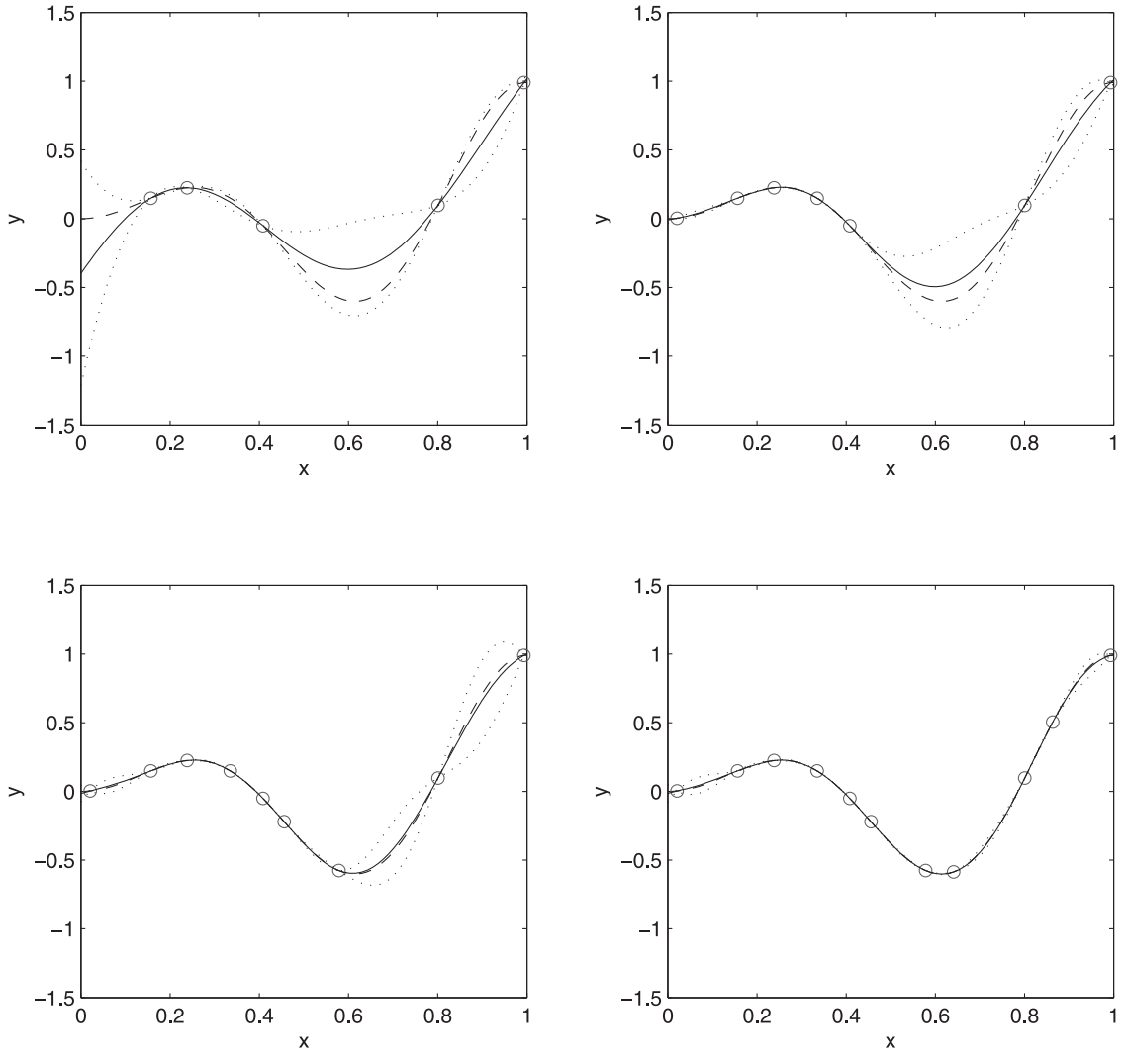

Figure 1. Illustration of a Gaussian process emulator. Dashed line: $y = x\sin 8x$; circles: points sampled from this function; solid line: mean emulator prediction; dotted lines: 95% prediction intervals.

2.3. Model Inadequacy

[21] Having replaced the simulator with an emulator, equation (1) can be rewritten as

$$\zeta(\upsilon) = \rho\eta(\upsilon, \theta) + \delta(\upsilon)\,. \tag{10}$$

[22] The model inadequacy $\delta(\upsilon)$ is also taken to be a Gaussian process of the form $\delta(\upsilon) \sim \mathcal{N}[m_\delta(\upsilon), c_\delta(\upsilon, \upsilon')]$.

[23] Note that a further coefficient ρ has been introduced in equation (10), which also has to be estimated (KOH2001). From a statistical point of view ρ can be thought of as a regression coefficient on the emulator, which tends to diminish if the data are better explained by the Gaussian process model inadequacy. In practice, the results of the analysis are easier to interpret from a physical point of view if ρ is fixed at unity.

[24] Furthermore, in equation (10) $\delta(\upsilon)$ is no longer the discrepancy between the simulator $S(\upsilon, \theta)$ and reality $\zeta(\upsilon)$ but is now the discrepancy between the emulator $\eta(\upsilon, \theta)$ (scaled by ρ) and reality $\zeta(\upsilon)$. If the emulator is a good one, then this change in the nature of $\delta(\upsilon)$ will not be significant, but if the number of training runs for the emulator is small (with regard to the dimensionality of the input space) or the simulator response is particularly complex, then the emulator may not be a close approximation to $S(\upsilon, \theta)$ and $\delta(\upsilon)$ will be called upon to compensate for deficiencies in the emulator as well as the simulator.

2.4. Calibration and Prediction

[25] The calibration data comprise r observations $z^\dagger = \left(z_1^\dagger, ..., z_r^\dagger\right)^{\mathrm{T}}$, where $z_i^\dagger$ is an observation of $\zeta(\upsilon_i)$ for known input variables $\upsilon_i^\dagger$, but subject to observation error. In addition, the computer code has been run at q points $\left(\upsilon_j, t_j\right)$ generating outputs $y = (y_1, \ldots, y_q)^{\mathrm{T}}$. The full set of data that are available for the analysis are $d^{\mathrm{T}} = (y^{\mathrm{T}}, z^{\dagger\mathrm{T}})$. Generally, q will be much greater than r, since even if the computer code is expensive to run it will still be much cheaper than obtaining field observations.

[26] There are the following distributions and parameters to be estimated: (1) the posterior distribution of the calibration parameters θ; (2) the regression coefficients β_1 and β_2

of the emulator $\eta(\cdot)$ and model inadequacy $\delta(\cdot)$, respectively; (3) the regression coefficient ρ; (4) the observation errors e_i, which are assumed to be independently distributed as $\mathcal{N}(0, \lambda)$ (where we take λ as the variance for consistency with the multivariate representation); and (5) the parameters σ^2 and ω_j of the covariance function for the emulator $\eta(\cdot)$ model inadequacy $\delta(\cdot)$, written as $\psi_1 = (\sigma_1^2, \omega_{1v_1}, ..., \omega_{1v_{nv}}, \omega_{1t_1}, ..., \omega_{1t_{nt}})$ and $\psi_2 = (\sigma_2^2, \omega_{2v_1}, ..., \omega_{2v_{nv}})$, respectively, where nv and nt are the numbers of variables in the vectors v and t, respectively.

[27] Here we demonstrate two approaches to this computation, first, adopting the assumptions of KOH2001, and second, adopting a rather more general approach and implementing the computation using the Markov chain Monte Carlo (MCMC) simulation.

2.4.1. Approach of KOH2001

[28] KOH2001 concede that integrating out the hyperparameters ρ, λ, ψ_1, and ψ_2 would be excessively burdensome, so integrate out β_1 and β_2 analytically under the assumption of a uniform distribution, and employ optimization to estimate fixed values of ρ, λ, ψ_1, and ψ_2. The posterior distribution of θ is then estimated to be conditional upon these parameter estimates. The prior distribution of θ is assumed to be multivariate normal. KOH2001 make the greatest possible use of the algebraic properties of the Gaussian process to write down expressions for the posterior distributions of θ, β, and ϕ given d. These expressions are not repeated here, and the reader is directed to *Kennedy and O'Hagan* [2001a, 2001b]. The equations have been implemented in the R statistical programming language [*Development Core Team*, 2005] in the package BACCO [*Hankin*, 2005], which was modified and extended by the authors to implement the examples described in this paper. The calibration proceeds in two stages. In the first stage the computer model output y is used to estimate ψ_1 (the hyperparameters of the emulator). In the second stage, ρ, λ, and ψ_2 (the hyperparameters of the Gaussian process representing model inadequacy) are estimated by maximizing $p(\rho, \lambda, \psi_2 | d, \psi_1)$. Finally, the posterior distribution of $p(\theta | d, \rho, \lambda, \psi_1, \psi_2)$ is computed.

2.4.2. MCMC

[29] A fully Bayesian analysis involves integrating out the hyperparameters ρ, λ, ψ_1, and ψ_2. This is not tractable algebraically, but can be achieved using MCMC, which yields a numerical approximation to the posterior probability distribution. The problem is made considerably simpler to solve by first analytically integrating out β_1 and β_2, which are often highly correlated, under the assumption of an improper uniform distribution, as described by *Kennedy and O'Hagan* [2001b] and *O'Hagan and Forster* [2004], and solving the problem conditionally on this assumption. An alternative would have been to follow the practice of *Higdon et al.* [2004] who scale the problem to remove the β parameters, effectively taking point estimates from preanalysis, and allowing the Gaussian processes to absorb the distributional uncertainty. The MCMC has been implemented using a random walk Metropolis-Hastings algorithm [*Gamerman*, 1997]. In order to ensure that convergence to the posterior probability distribution was achieved, the chains were monitored, both visually and using the *gibbsit* routine of *Raftery and Lewis* [1996], and the calculation was repeated using a number of starting locations [*Gelman*, 1996].

[30] In practice, the incorporation of the contribution of $\delta(v)$ in calibrated prediction (equation (5)) is slightly different, depending on the approach taken to the calibration. In the case of the KOH2001 approach, $p(\theta | d, \rho, \lambda, \psi_1, \psi_2)$ is integrated out and the (nonparametric) posterior distribution of θ is sampled using the Metropolis-Hastings algorithm. The predictive distribution can then be estimated given each realization of θ, and Monte Carlo estimates can be made of the expectation and quantiles of the predictive distribution. In the case of the MCMC approach, the sample is taken from the full distribution $p(\theta, d, \rho, \lambda, \psi_1, \psi_2)$, and the predictive distribution is estimated from the entire converged Markov chain.

3. A Synthetic Example

[31] To illustrate the proposed approach, a simple synthetic example is first presented. Synthetic observations have been generated from "reality" which enacts the function, $z = \exp(x) - 1$ but this is taken as being unknown. Cases have been tested with 10 and 20 "observations" of this function on a regular grid of 5 points, i.e., two or four observations at each observation location. The observation error, which is also taken as being unknown, is uncorrelated and simulated from $e \sim \mathcal{N}(0, 0.05^2)$. The computer code that is to be calibrated to these observations is taken as being a "black box" though, in fact, it implements the function $S(x, \theta) = \theta x^2$, i.e., it has one calibration parameter θ. It is recognized that θ is uncertain and, furthermore, that the computer code is inevitably an imperfect representation of reality, as it obviously is in this case. The Bayesian calibration procedure reflects both of these sources of uncertainty.

[32] In this example the basis for the code has been taken as $h(x, \theta) = (1, x, \theta)$ and for the model inadequacy, $h(x) = x$. The first step in the calibrated prediction is to fit an emulator to the "code runs." This has been done using a Latin hypercube sample of 20 points on [0,1] $\times$ [1,3], with priors $\sigma_1^2 \sim \mathcal{N}(0.1, 0.3^2), \omega_{1x} \sim \mathcal{N}(1, 1)$, and $\omega_{1t} \sim \mathcal{N}(1, 1)$. Note that there is no prior specified for β_1, as this is estimated indirectly from the posterior values of the other variables, under the assumption of an improper uniform prior. The prior on σ_1 is estimated from the fit of a linear model to the code runs. Priors on ω_{1x} and ω_{1t} represent beliefs about the rate of variation (described in the computer code analysis literature as "roughness") with v and t, respectively, expected from the model output. These priors are estimated using the method of *Oakley* [2002], namely, by visual examination of Gaussian processes with different parameter values, in order to identify a plausible range of ω_{1x} and ω_{1t}.

[33] In the following, the emulation and calibration results calculated in BACCO are reported, though the results in MCMC are very similar in the case of this first example. The following parameter values are estimated: $\hat{\beta}_1 = (-0.725, 2.2, 0.757)^T$, $\hat{\sigma}_1^2 = 0.902$, $\hat{\omega}_{1x} = 1.118$, and $\hat{\omega}_{1t} = 0.117$. The fitted emulator is illustrated by the contours in Figure 2a, while the training points are illustrated with points. Figure 2b shows the 95% error range compared with the exact function at a number of validation points that were not used for training, marked in Figure 2a by circles. It can be seen that both the mean error and the

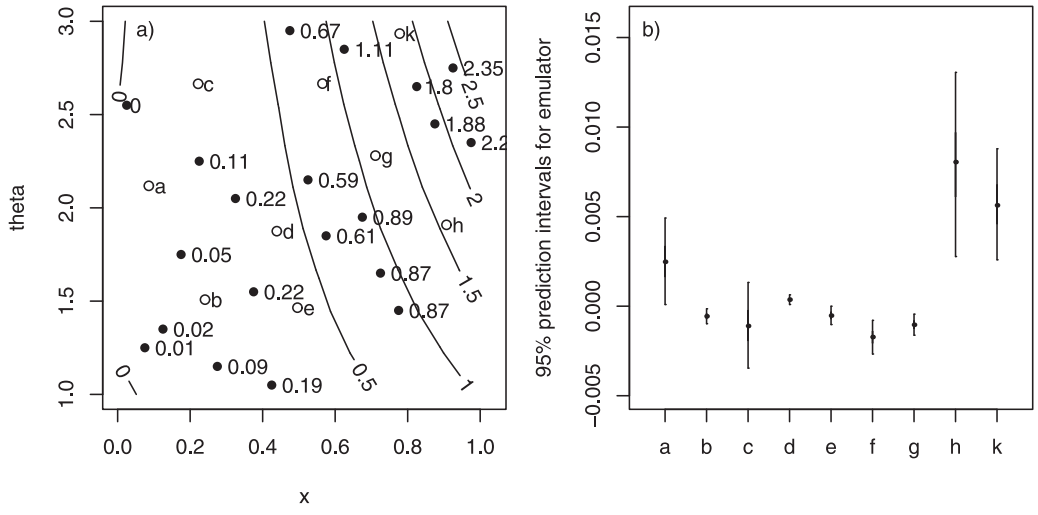

Figure 2. Testing the code emulator: (a) contour map of the emulator response surface; solid points are where the "model" has been run, circles are where the emulator accuracy has been tested. (b) Ninety five percent prediction interval for the emulator at locations labeled in Figure 2a (central points are true values).

uncertainty in the emulator are low, but are higher further away from data points, particularly close to the edge of the domain where the prediction intervals expand rapidly.

[34] Next, the observations are used in calibrated prediction. The prior distribution on θ is taken as $\theta \sim \mathcal{N}\left(1.94, 0.71^2\right)$, the prior distribution of the observation error e is taken as $e \sim \mathcal{N}(0, 0.05^2)$, and ρ is fixed at unity. With 10 observations, posterior distribution of θ is illustrated in Figure 3 and the following parameter values are estimated $\hat{\beta}_2 = 0.394, \hat{\sigma}_2^2 = 0.15$, and $\hat{\omega}_{2x} = 3.43$. The posterior standard deviation distribution $\sqrt{\lambda}$ of the observation error was estimated to be 0.066, compared with the value of 0.05 which was used to simulate the data.

[35] The predictive distribution is illustrated in Figure 4. Also illustrated are the unknown "reality" and the best fit that is obtained by tuning θ in the computer model, using an ordinary least squares approach. Because of the inadequacies in this computer model (it is the wrong function!) there is a limit to how well it can predict reality. The least squares estimate of $\theta = 1.960$ with a standard error of 0.0811. In the case where 20 observations were taken, the estimated posterior standard deviation $\sqrt{\lambda}$ of the observation error is smaller, at 0.059, and the variance and roughness of the discrepancy function are slightly larger.

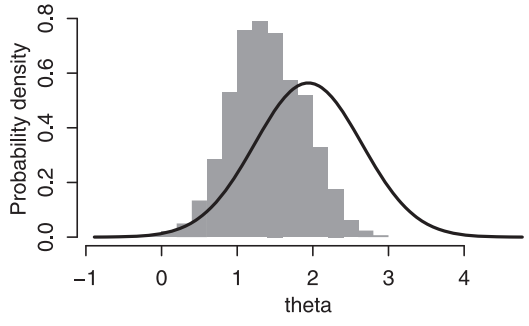

Figure 3. Prior (line) and posterior (histogram) distributions of θ.

[36] In both cases the mean of the calibrated prediction is a much better prediction of reality than the best "tuned" model. Divergence of the prediction limits away from the observations is evident. The width of prediction limits also reduces with an increasing number of observations, especially within the range of observations. The effect of increasing the number of observations on the uncertainty in extrapolation is less evident. Additional observations also yield little reduction in the estimate of observation error standard deviation, since this was already well-estimated. The prediction limits bound reality. Recall that these are predictions of reality, not of observations, and thus the bounds are narrower than the data when, as in this case, there is appreciable observation error. Consequently, the bounds are not expected to bound 90% of the data.

[37] The sensitivity of the calibrated prediction to the choice of regression bases for the emulator and the model inadequacy function have been investigated. Within the range of the data (either code outputs in emulator generation, or measured data points), the Gaussian process follows the data, and its mean and variance are unaffected by the regression basis. However, in extrapolation (both in the emulator and in the model inadequacy), the Gaussian process follows the regression on the basis functions, deviating from the trend of the data at a rate depending on the "roughness" coefficient, ω. For example, if in the above example the model inadequacy regression basis is taken as x^3 instead of x, the calibrated predictions within the data range are almost identical, but in extrapolation, the calibrated mean departs rapidly from reality, while the prediction limits increase rapidly. Similarly, if the emulator regression basis is taken as 1 instead of $(1, x, \theta)$, the calibrated mean is approximately linear for large x, but not continuing the trend of the data; whereas if the basis is chosen to be $(1, x^2, \theta)$, the calibrated mean follows the code more closely.

[38] Using MCMC it is possible to circumvent the emulation step and call the simulator directly during the calibration procedure, which is feasible because the simulator

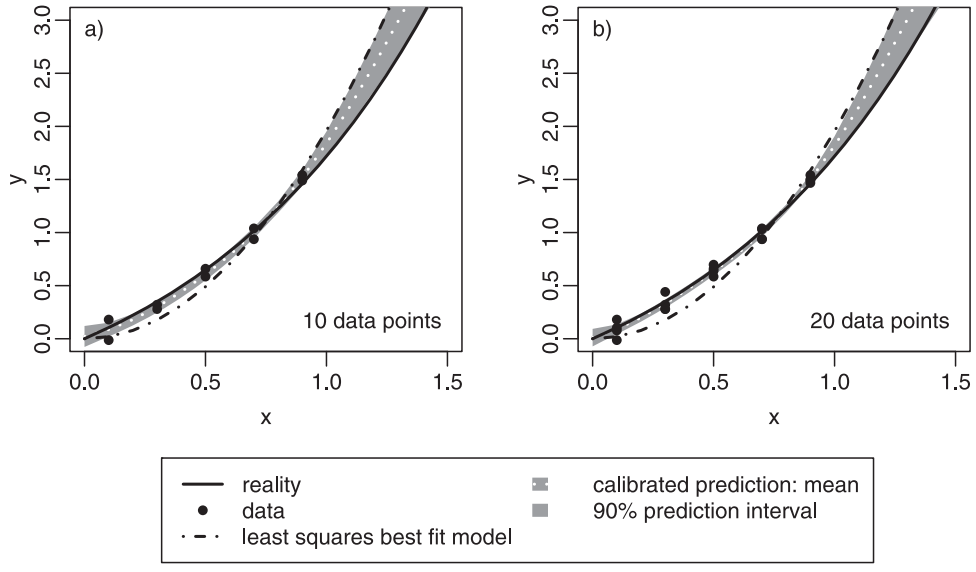

Figure 4. Calibrated prediction for (a) 10 "observation" points and (b) 20 "observation" points. The "least squares best model fit" is the best fit of the inadequate model to the data.

function is so simple. Doing so does not yield any noticeable improvement within the range where the emulator was trained. Avoiding emulation does, however, narrow the uncertainty in extrapolation, in which case there is no longer any need for an appropriate choice of emulator basis function. Prediction still depends upon the model inadequacy basis, the choice of which is significant when extrapolating beyond the observations, for the reasons explained above.

4. Flood Inundation Model Calibration

[39] *Aronica et al.* [2005] conducted an uncertainty analysis of the LISFLOOD-FP model [*Bates and De Roo*, 2000], using the GLUE methodology [*Beven and Binley*, 1992] and synthetic aperture radar observations of floodplain inundation. The same model and data set are used here to illustrate the practical application to a flood model of the Bayesian calibration methodology proposed herein. The approach could readily be applied to a more sophisticated hydrodynamic model, but LISFLOOD-FP has been adopted here to enable comparison with the previous uncertainty analysis.

4.1. Model Description and Application

[40] LISFLOOD is flood inundation model in which channel flow is handled using kinematic or diffusive versions of the one-dimensional St. Venant equations, while floodplain flow has a simple two-dimensional representation on a regular grid. The channel parameters required to run the model are its width (assuming a rectangular channel cross-section), bed slope, depth, and Manning's n value. Floodplain flows are discretized over a grid of square cells. It is assumed that the flow between two cells is simply a function of the free surface height difference between those cells. While this approach does not accurately represent the full hydrodynamics of floodplain flow, it is computationally simple and has been shown to give very similar results to a more accurate methods [*Horritt and Bates*, 2001, 2002].

[41] Application of the LISFLOOD model to a reach of the upper Thames near Buscot in Oxfordshire, UK was described by *Aronica et al.* [2005]. The same site has been used in the current analysis. The river at the site drains a catchment of 1000 km^2 and has a bankfull discharge of roughly 40 m^3/s. The test reach is bounded upstream by a gauged weir at Buscot (which provided the upstream boundary condition). Flows are reasonably well-confined at the downstream end of the site. The floodplain topography was obtained from stereophotogrammetry at a 50 m scale with a vertical accuracy of $\pm$ 25 cm.

[42] Calibration data for the case study were available in the form of a SAR observation from the ERS-1 satellite, whose overpass coincided with a flood in December 1992. The flood had a return period of approximately 5 years and a peak discharge of 76 m^3/s. The discharge at the time of the satellite overpass had subsided to 73 m^3/s. The SAR image provided a map of inundation extent with boundaries accurate to $\pm$ 50 m (Figure 5). The broadness of the hydrograph along the short length of reach means that steady state boundary conditions are as successful as dynamically varying boundary conditions; so for the application

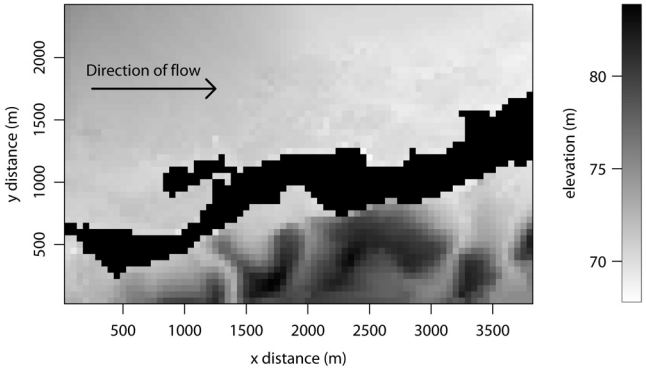

Figure 5. Floodplain topography at Buscot with SAR image of flood outline superimposed (in black).

reported here LISFLOOD has been run using steady state boundary conditions at a time step of 1 s. While this time step is small, the computational efficiency of the code means that each simulation still took less than 1 min on a standard PC.

[43] *Hall et al.* [2005] conducted a comprehensive sensitivity analysis of the LISFLOOD model at the Buscot site, concluding that the model is quite insensitive to floodplain friction parameterization. The main uncertain variable that significantly influences flood depth predictions is the channel friction parameterization using Manning's n, which is therefore singled out for analysis in the following.

4.2. Obtaining Flood Depths From Flood Outlines

[44] The primary predictive variable for flood risk analysis is water surface elevation, which is combined with land elevation to obtain flood depth. Water surface elevation is also a fundamental predictive variable in hydraulic modeling of rivers, so is an appropriate observable to use for calibration purposes. However, the spatial observation data at the Buscot site were of flood outline, so a procedure is required to convert flood outlines into water surface elevations. A solution to this problem in the context of satellite SAR imagery has been discussed by *Oberstadler et al.* [1997]. Simply superimposing the flood outline on the digital elevation model (DEM) of the site and extracting elevations at the edge of the flood outline can yield a wide range of elevations for each pixel at the edge of the floodplain because of the coarse resolution of the SAR image. *Schumann et al.* [2007] proposed a regression and elevation-based flood information extraction (REFIX) model that applies linear, piecewise linear or nonlinear regression modeling to the extracted water heights. Here a similar approach has been applied by locally matching patches from a large database, obtained by *Hall et al.* [2005], of 638 simulated water surfaces, by varying discharges, roughness, and channel geometries. The database was searched in order to find the water surfaces that exactly matched the wet/dry classification of the nine grid cells centered on the cell of interest. The mean water surface elevation in the cell of interest of these locally matched water surfaces was taken as the observed water surface elevation. This yields water surface elevation estimates along the two edges of the flooded area. The water surface elevations inferred from this method are plotted in Figure 6a, and the spatial coordinates of the locations at which these points have been extracted are plotted in Figure 6b. For a gently varying topography of the type at this site, the water surface elevation is expected to vary fairly smoothly down the floodplain. The outliers in the data set illustrated in Figure 6a can be attributed to misclassification of the flooded area in the processing of the SAR image.

4.3. Constructing the Emulator of the LISFLOOD Code

[45] The LISFLOOD code is treated as being a black box $z = S(x, y, n)$, which generates predictions of flood depth z for given values of Manning's n and locations x and y. All other input variables (e.g., channel and floodplain geometry and upstream discharge) are taken as being fixed. The emulator could, of course, be extended to include other inputs, such as discharge, if these are to be taken as variables either in calibration or in calibrated prediction. Note that while the prediction z is taken as a scalar output of $\eta(x, y, n)$, in practice a spatial field of depths is generated by using the index variables x and y. This provides a convenient mechanism for producing spatial fields of outputs from a univariate emulator.

[46] From an ensemble of runs of the LISFLOOD model at different values of n a small, stratified sample of runs was selected that efficiently spanned the input and output space of the model. The existing database of 638 model outputs was used to identify the range of model predictions and the selected runs were optimized to span the coverage of this range, minimizing the distance of any point in the space from the training run. While this approach did rely upon having a relatively large database of model runs, the emulator construction is still much more efficient than calling the simulator directly during the calibration procedure, which could require tens of thousands of calls to the simulator. Moreover, it does not rely upon running the code at specific points, provided the available database spans the plausible range of input data.

[47] Figure 6c illustrates the water surface profiles obtained from these selected runs, which span reasonably well the observations shown in Figure 6a. A random sampling of 40 water surface elevations at points on the boundaries of the flood outline were extracted from these selected runs. The projection of these points onto $x \times y$ is illustrated in Figure 6d. Note that these points are concentrated near the boundaries of the flooded area rather than being uniformly distributed over $x \times y$, as it is here that the emulator is required to be most accurate in order to make best use of the observation data and also to generate accurate flood predictions.

[48] A parsimonious selection of points to train the emulator is necessary as the KOH2001 calibration procedure involves inverting a matrix whose dimension is the same as the number of training points plus the number of observations. A careless choice of points can result in the matrix being ill-conditioned. Furthermore, it is customary to rescale all variables to [0,1]. This enables evaluation of the "roughness" parameters ψ_1 of the Gaussian process.

[49] The Gaussian process emulator was established with the regression basis $h() = (1, x, n)$, y having been omitted as its regression coefficient β_{1y} was not significant. A combination of Nelder-Mead and simulated annealing optimization was used to estimate hyperparameters of the emulator, with vague priors. The following parameter values are estimated: $\hat{\sigma}_1^2 = 0.011$, $\hat{\omega}_{1x} = 5.62$, $\hat{\omega}_{1y} = 19.2$, and $\hat{\omega}_{1n} = 3.94$, and the coefficients for β_{1y} were 0.58 (constant), $-0.78(x)$, $0.46(n)$. Figure 7 shows the mean and standard deviation of the difference between the emulator and the output from LISFLOOD runs not used to train the emulator, for points close to, midway between, and outside the range of the training runs. The distance between the mean emulator prediction and the verification points is greater in extrapolation. It is within 0.1 m for interpolation near training points, with the exception of some localized irregularities. These are explained by local irregularities in the water surface from the runs of the LISFLOOD model with which the emulator is compared. Because of the choice of covariance function, the emulator generates a smooth continuous surface, so it does not do well at reproducing sudden jumps. Paradoxically, at a site like Buscot, these sudden jumps are rather unrealistic, so the smoothing

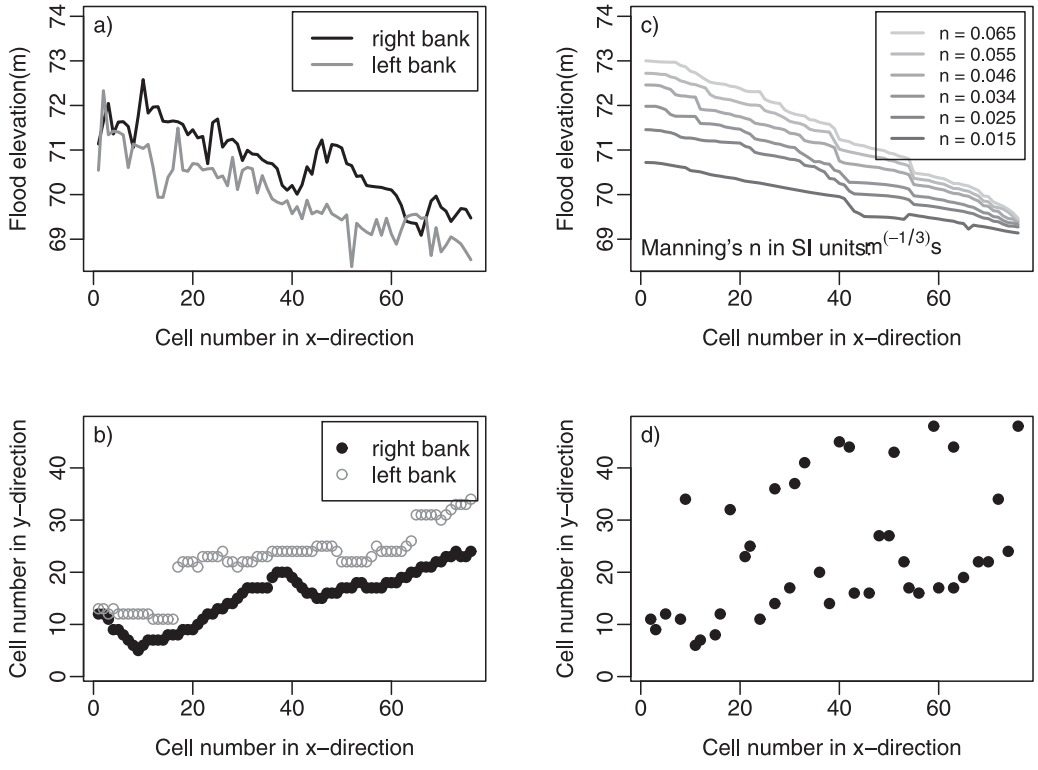

Figure 6. LISFLOOD calibration and emulation data. (a) Flood depths inferred from the SAR flood outline image. (b) Points where flood depth values were extracted from SAR observations. (c) Typical LISFLOOD water surface profiles at different values of Manning's n. (d) Points used to construct the LISFLOOD emulator.

applied by the emulator is actually rather beneficial, even if it means that locally the emulator validation is not perfect when compared with some LISFLOOD runs. Figure 7 also illustrates how the standard deviation of the emulator predictions increases further from the training points, and in particular in extrapolation (as was also seen in the synthetic example presented above).

[50] The Gaussian process assumption in the emulator may be tested by examining the absolute differences $g = |m^*(x, y, n) - S(x, y, n)|$ between the posterior mean emulator prediction $m^*(x, y, n)$ (see equation (9)) and the output $S(x, y, n)$ from LISFLOOD at points that were not used to train the emulator. This difference g was computed at a total of 2037 points on the flooded surface from 486

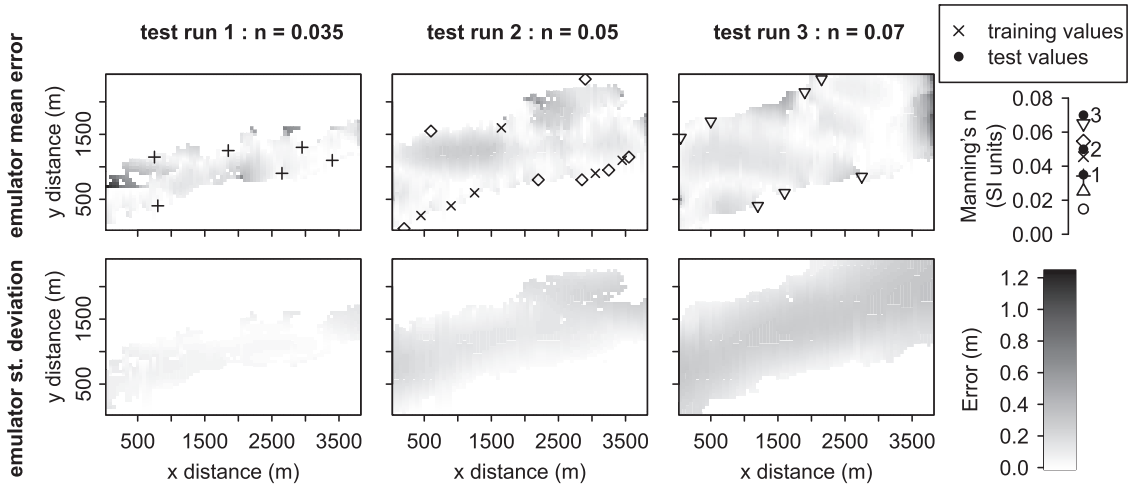

Figure 7. Mean emulator prediction and standard deviation of the predicted emulator uncertainty for three different values of Manning's n: close to, in between, and away from the training runs. Positions of nearby training points and values of test and training runs are illustrated.

validation runs and written as $g = (g_1, \ldots, g_{2037})^{\mathrm{T}}$. The expression for evaluation of the posterior emulator covariance matrix $\mathbf{A}$ for these validation points is given by *Oakley and O'Hagan* [2002]. While equation (9) describes the posterior emulator distribution at a point, here we are concerned with the distribution at a sample of points, so the variance term $c^*(x, x)$ is replaced by a matrix $\mathbf{Q}^{\mathrm{T}}$ such that $\mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \mathbf{A}$ so that

$$\hat{\sigma}_1^{-1}\mathbf{Q}^{-\mathrm{T}}\left(\frac{q-p-2}{q-p}\right)^{-0.5} g \sim t_{q-p}\,. \tag{11}$$

[51] Here the number of simulator outputs used to train the emulator $q = 6$, while $p = 3$, so the degrees of freedom of the t distribution will be 3. Thus, a test of the assumed emulator distribution is that the vector of validation differences g, when transformed according to equation (11) should be t_3 distributed. The results of this test, based upon our 2037 validation outputs, and achieved through eigenvector transformation using singular value decomposition are presented in Figure 8. The transformed differences have mean 0.051, standard deviation 1.877 and can be compared visually with the t_3 distribution that is superimposed on Figure 8, showing good agreement. The result is not sensitive to the values of σ and ω, though for small ω the correlation matrix $\mathbf{A}$ becomes particularly ill-conditioned.

4.4. Calibration

[52] The attraction of Bayesian calibration is that it enables the incorporation of prior knowledge about uncertain physical quantities. For example, river modelers and engineers can often generate reasonable estimates of Manning's n from observations and measurements of the river channel [*Engman*, 1986; *Arcement and Schneider*, 1990; *Yen*, 1992]. Indeed, it has been argued that modelers should not depart from these physical observations [*Cunge*, 2003]. However, it is clear that quantities like Manning's n may vary with time (because of changes in vegetation or river morphology) and are scale dependent, so they cannot be determined precisely through point observation, though observation should be a guide to the physically reasonable range of variation, encoded as a prior distribution. For the site in question a truncated Gaussian prior for Manning's n was taken as $n \sim \mathcal{N}(0.0265, 0.0224^2)$, subject to $n > 0$. The prior distribution of the variance λ of the observation errors was taken as being lognormally distributed $\ln(\lambda) \sim \mathcal{N}(-6.25, 0.01)$ so that the mean of $\sqrt{\lambda}$ is 0.15 m, though a value of 0.25 m was also tried. Recall that within the Bayesian framework adopted here the observation error term represents the difference between observations and reality and should not be used to represent other model uncertainties, as can be the case in less well-structured inference frameworks.

[53] In the calibration step these prior estimates were updated using a sample of 26 points from the flood depth observations (Figure 6a) in order to generate a posterior distribution of Manning's n (Figure 9), and a model inadequacy function. Eight hundred iterations of a combination of Nelder-Mead and simulated annealing optimization were used to estimate hyperparameters of the model inadequacy as follows: $\hat{\sigma}_2^2 = 0.0092$, $\hat{\omega}_{2x} = 4.96$, $\hat{\omega}_{2x} = 15.4$. We observe that the posterior distribution of Manning's n is less diffuse than the prior and is shifted slightly to the right.

4.5. Implementation on BACCO and MCMC

[54] The foregoing results have been achieved using BACCO. However, a comparison between these results and those achieved using MCMC provided further methodological insight. While the proposal of KOH2001 to use optimization to find point estimates of the Gaussian process hyperparameters was motivated by the desire to increase the size of problem amenable to this method, it was found that in practice for this problem, the BACCO optimization failed when more than 26 data points were used; whereas a solution with MCMC was achieved with up to 93 data points, with both cases using 40 training points for the emulator. Moreover, the MCMC method showed greater sensitivity to a poor choice of emulator training points, as it allowed a wider choice of prior distributions to be explored. In other examples, this greater sensitivity of MCMC has highlighted the implications of poor choice of correlation structure or of basis functions for the Gaussian processes describing the emulator and model inadequacy, recognized by slow convergence of individual variables or by excessive dependence of the marginal distribution of the affected parameters or hyperparameters on their priors.

[55] Statistical authors working in the field of Bayesian calibration of computer models differ in their preference between the semianalytical approach encoded in BACCO and fully MCMC approaches. Experiences in this study have revealed the following insights.

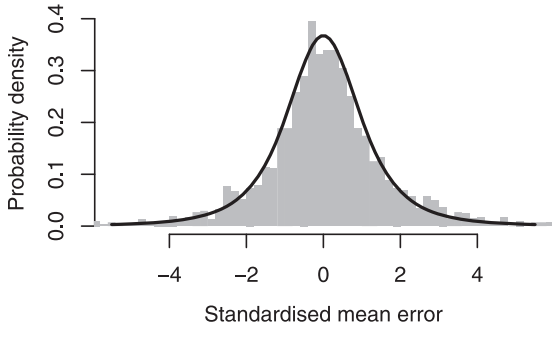

Figure 8. Histogram of standardized residuals between the mean emulator prediction and the LISFLOOD model output at validation points. Continuous line is the pdf of a t distribution with 3 degrees of freedom.

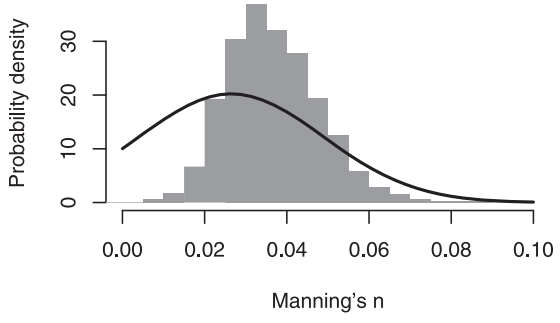

Figure 9. Prior (line) and posterior (histogram) distributions of Manning's n.

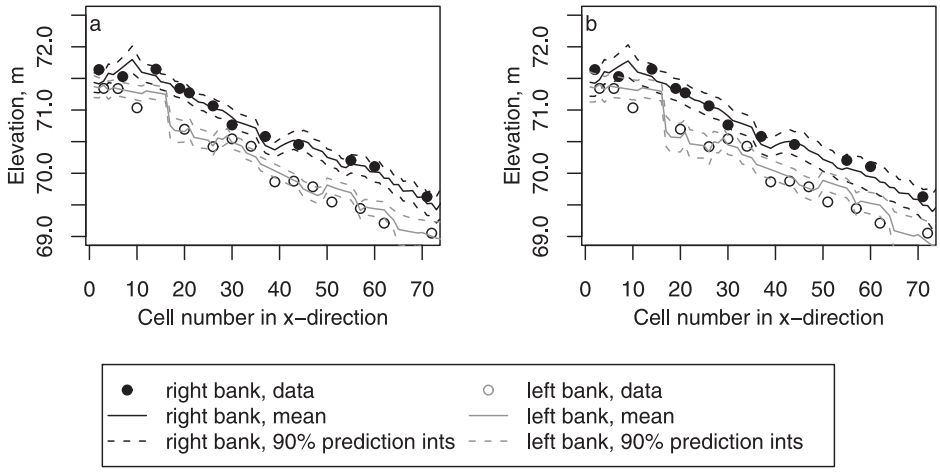

Figure 10. Calibrated prediction of flood depths on the right and left bank of the flooded area. (a) Prior mean for the observation error variance of $\lambda = (0.15\ \text{m})^2$. (b) Prior mean for the observation error variance of $\lambda = (0.25\ \text{m})^2$.

[56] 1. The greater diagnostic power of the MCMC method lies in the fact that it estimates the entire distribution of all uncertain parameters, whereas the optimization in BACCO only provides a point estimate. A flat or multimodal output distribution is more informative than a single maximum.

[57] 2. It is necessary to employ an emulator for the code in BACCO, even if, as in the mock example described in section 3, the code is sufficiently simple so that there is no computational penalty in using it directly. As already noted, while in interpolation the emulator is accurate and makes negligible contribution to the predictive uncertainty, in extrapolation the uncertainties are large and the sensitivity to choice of emulator basis functions is significant.

[58] 3. In BACCO, the prior on the parameters θ must be (multivariate) normal. In addition, it is assumed that the covariance structure for the code emulator is a negative-squared exponential of the distance, in x and θ, between the code output values. More flexible assumptions are admissible within an MCMC implementation.

[59] Taken together, these assumptions do restrict the applications that can be treated using BACCO, preventing, for example, the analysis of a situation involving discontinuities.

4.6. Calibrated Prediction

[60] Calibrated prediction involves computing the integral in equation (5) for each point in the floodplain. This is implemented numerically by sampling from the posterior distribution of n and combining this with posterior estimates of the model inadequacy to generate a Monte Carlo sample of predicted flood elevations, from which a predictive distribution can be estimated (Figure 10). These predictive flood elevations are then superimposed on the DEM to obtain predictive flood depths (Figure 11).

[61] The mean water surface elevation shown in Figure 10 is less regular than might be expected, but note that this corresponds to the vector of points down each bank (Figure 6b) rather than a smooth river centerline. Figure 10 also shows the mean water surface elevation that is predicted by repeating the calibration using a prior mean for the observation error variance of $\lambda = (0.25\ \text{m})^2$, as opposed to $(0.15\ \text{m})^2$. In both cases the prior distribution was specified quite tightly. The lower observation error variance prior leads to smaller variance on the other uncertain terms and, consequently, a more precise prediction. Choice of a more diffuse prior on the observation error may lead to an implausible proportion of the data variability being attributed to model inadequacy. Choice of a more precise prior for the "roughness" coefficients, ω, does not appear to affect this.

[62] The result achieved using the $\lambda = (0.15\ \text{m})^2$ prior observation error variance can be compared with the flood likelihood map for the same event generated by *Aronica et al.* [2005] who used GLUE (Figure 12). The images are similar, although some difference may be expected from the difference in preparation of the data: the *Aronica et al.* [2005] analysis predicted binary flood inundation on a 50 m grid, whereas the analysis reported in this paper has been performed in terms of depths, making it more likely to show small areas of isolated inundation. Moreover, *Aronica et al.*

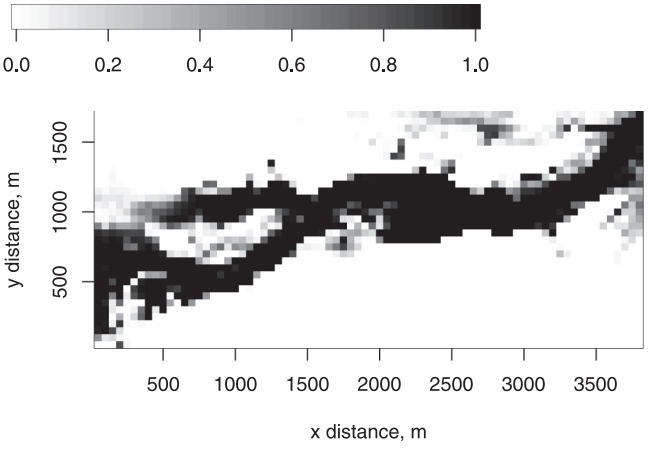

Figure 11. Map of the calibrated predictive probability of inundation for the December 1992 event for the river Thames.

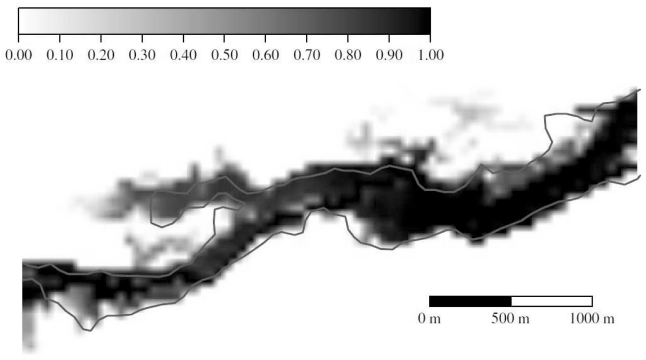

Figure 12. Map of likelihood of inundation for the December 1992 event for the river Thames obtained by *Aronica et al.* [2005] using GLUE.

[2005] noted the sensitivity of the GLUE prediction to the choice of threshold for "behavioral" runs, a deficiency also observed by *Montanari* [2005]. While we have illustrated that the Bayesian predictions are sensitive to the prior distribution of observation error, we argue that it is more natural to express a prior belief in the distribution of observation error (which is the physical consequence of the quality of the observation process) than in the arbitrary behavioral/nonbehavioral threshold used in GLUE. Moreover, the results generated here can strictly be referred to as predictive probabilities of flooding, while the results presented by *Aronica et al.* [2005] are probabilities in a rather informal sense.

[63] 1. The likelihood function is formal in the sense of being the probability of observations given the model, rather than the scoring function employed by *Aronica et al.* [2005]. As *Mantovan and Todini* [2006] have demonstrated, the use of informal likelihood functions is incoherent.

[64] 2. Observation error is explicitly included in the analysis and is excluded from the prediction, so that the predictive probabilities are of flood depths without observation error.

[65] 3. Prior knowledge of channel roughness and observation accuracy has been formally incorporated in the analysis via prior distributions.

[66] 4. A model discrepancy function has been estimated, which reflects the inevitable inadequacies in the LISFLOOD model.

[67] 5. The probability distribution plotted in Figure 11 is predictive in the sense of equation (5) which, as *Todini* [2008] has also argued, is the only distribution of relevance to decision-makers.

4.7. Extension to Variable and Uncertain Discharge

[68] As in the example presented by *Aronica et al.* [2005], the example presented here is based upon the prediction of flood probabilities for a given inflow. The more general risk analysis problem is for prediction of flooding probabilities where the long run discharge q_f (where the subscript "f" denotes "future") is specified as a distribution $f(q_f)$ obtained from the analysis of flood frequency. In this case, the simulator $S(x, y, q, n)$ is a function of the inflow q as well as the (x, y) coordinates and n. The model inadequacy δ might also be expected to vary as function of q, though to estimate this function will require that $z^\dagger$ includes observations at a range of values of $q^\dagger$. If $q^\dagger$ is taken to be error-free, then the predictive distribution is written

$$P\left[\zeta(x,y) \in \mathcal{Z}|z^\dagger\right] = c \iiiint I_{\mathcal{Z}}\left[S(x,y,q,n) + \delta_{xyq}\right] f\left(z^\dagger|n,\ \delta_{xyq},\ e\right) f(n,\ \delta_{xyq},\ e)\, f(q_f)\, dn d\delta_{xyq} de dq\,. \tag{12}$$

[69] If the discharge $q^\dagger$ includes observation errors, then the flow q_0 at the time of the calibration events is included in the joint prior distribution, $f\left(n, \delta_{xyq}, e, q_0\right)$ which can be updated in the calibration process to yield the posterior distribution $f\left(n, \delta_v, e, q_0|z^\dagger\right)$. In prediction, q_0 is dropped from the joint distribution and replaced with $f(q_f)$ to yield the following predictive distribution

$$P\left[\zeta(x,y) \in \mathcal{Z}|z^\dagger\right] = c \iiiint I_{\mathcal{Z}}\left[S(x,y,q,n) + \delta_{xyq}\right] f\left(z^\dagger|n,\ \delta_{xyq},\ e, q_0\right) f(n,\ \delta_{xyq},\ e)\, f(q_f)\, dn d\delta_{xyq} de dq\,. \tag{13}$$

[70] The notion that q_0 is only known to within some tolerance during a flood event, and that beliefs about q_0 may be updated during a calibration process is consistent with flood modeling practice; whereby modelers may question gauged measurements in extreme flows, especially where those estimates yield flood depths in the model domain that are not consistent with observations. The Bayesian approach provides a formal framework for this process of modifying beliefs about gauged flows given observations of flood depths or extents.

5. Conclusions

[71] The Bayesian formulation of the flood model calibration problem, as presented in this paper, is a complete and coherent description of the uncertainties in calibration parameters, observation errors, and computer model inadequacy. The incorporation of a model discrepancy function to represent the distance between best model predictions and reality is a particularly significant element within the framework and avoids compensation for model structural uncertainty in the posterior distribution of the model parameters. The use of a Gaussian process emulator of the computer model provides the possibility of using Bayesian calibration for computationally expensive hydrodynamic models.

[72] Bayesian calibration presents considerable challenges, as pointed out by *Beven et al.* [2007], including the correct specification of the likelihood function and treatment of heteroscedastic errors. The approach demonstrated here is based largely upon the use of Gaussian processes, but this does not imply that all of the distributions under consideration are taken as Gaussian. In particular, the posterior distribution of the model parameters is nonparametric and may take any distributional form. Elsewhere, Gaussian assumptions may be circumvented to some extent through judicious use of transformations. The advantage of employing Gaussian processes is that they provide considerable flexibility in the mean and covariance functions and enable analytical solutions to at least some of the integrals in the Bayesian updating. The use of an additive Gaussian process

model inadequacy might be challenged in particular, but, unlike naive additive formulations (e.g., the addition of independent and identically distributed error), in KOH2001 the additive term is general enough to represent in a flexible way (subject to modest continuity assumption) model inadequacy that is evident from observations. The model inadequacy function can be further extended, in physically realistic ways, by incorporating sufficient explanatory variables (e.g., discharge, spatial location, and antecedent conditions) in the model inadequacy function; in the example presented here we have incorporated location in two-dimensional space in the inadequacy function. As the model inadequacy function becomes more elaborate (and perhaps also the number of calibration parameters in the computer model increases) then the Bayesian calibration problem will suffer from identifiability problems unless sufficiently precise priors are available. The optimal complexity of the computer model and surrounding statistical framework will have to be decided on a case-by-case basis, supported with systematic sensitivity analysis. However, as the examples in this paper have illustrated, even with rather scarce data meaningful inference is feasible without requiring untenable prior assumptions.

[73] Two approaches to implementing the Bayesian updating have been compared: (1) a direct solution of *Kennedy and O'Hagan*'s [2001a, 2001b] equations using Hankin's [*Aronica et al.*, 2005] R routines called BACCO; and (2) a more general computational approach using MCMC. A direct solution makes the most of the computational opportunities that are offered by a reliance upon Gaussian processes, but in practice MCMC can be faster and more flexible, for example, in the choice of priors. The greater sensitivity of the MCMC to poor problem setup is a useful diagnostic.

[74] We have explored the implications of critical methodological choices, including the use of an emulator, specification of prior probability distributions, and specification of the basis functions for the Gaussian processes. In common with previous investigations, we have demonstrated that the specification of basis functions is not significant provided observations span the range of prediction. In extrapolation, the effect of basis functions becomes more significant.

[75] We have examined the Gaussian process assumption in the emulator and identified that, in the context of the numerical model simulations tested here, the emulator is unbiased but the variance in the difference between a sample of validation points and the emulator mean function is greater than the Gaussian process assumption would imply. The test presented here provides a route to assessing the Gaussian process assumption when validation runs are available; a topic that merits future investigation in future.

[76] While the use of an emulator for the computer simulator is deeply embedded in *Kennedy and O'Hagan*'s [2001a, 2001b] approach, in MCMC it is easy to circumvent the emulator. If the use of an emulator can be avoided, then the sensitivity in extrapolation to the choice of basis functions is removed, and predictive uncertainties are reduced. However, avoiding the emulation step and calling the simulation model directly requires a numerical approach to the calibration procedure, implemented here using MCMC. Convergence of the Markov chain may require tens of thousands of (serial) model runs, and so for computationally expensive simulation models would require prohibitively long run times.

[77] The methodology has been applied to calibration of a steady state two-dimensional flood inundation model using a synthetic aperture radar observation of flood outline. This has involved converting binary spatial observations to flood depths and construction of an emulator of the flood model. The results have been compared with previously published results based upon the same data set and flood inundation model, where the uncertainty was analyzed using GLUE. While the results from GLUE are known to be sensitive to the choice of "behavioral" threshold, the results from the Bayesian analysis also show sensitivity to the specification of prior distributions. However, we argue that these prior distributions relate to quantities for which there may be prior empirical data (for example, relating to the accuracy of observations) or about which experts may be expected to have well-formed prior beliefs (for example, relating to the range of plausible values of Manning's n for a given channel).

[78] **Acknowledgments.** The research described in this paper was funded in part by the UK Flood Risk Management Research Consortium under EPSRC grant EP/F020511.

References

Apel, H., B. Merz, and A. H. Thieken (2008), Quantification of uncertainties in flood risk assessments, *J. River Basin Manage.*, *6*, 149–162.

Arcement, G. J., Jr., and V. R. Schneider (1990), Guide for selecting Manning's roughness coefficients for natural channels and flood plains, *U. S. Geol. Surv., Water Suppl. Pap.*, *2330*.

Aronica, G., P. D. Bates, and M. S. Horritt (2005), Assessing the uncertainty in distributed model predictions using observed binary pattern information within GLUE, *Hydrol. Processes*, *16*, 2001–2016.

Bates, P. D., and A. P. J. De Roo (2000), A simple raster-based model for floodplain inundation, *J. Hydrol.*, *236*, 54–77.

Beven, K. (2006), A manifesto for the equifinality thesis, *J. Hydrol.*, *320*, 18–36.

Beven, K., P. Smith, and J. Freer (2007), Comment on "Hydrological forecasting uncertainty assessment: Incoherence of the GLUE methodology" by Pietro Mantovan and Ezio Todini, *J. Hydrol.*, *338*, 315–318.

Beven, K. J., and A. M. Binley (1992), The future of distributed models: Model calibration and uncertainty prediction, *Hydrol. Processes*, *6*, 279–298.

Campbell, K. (2006), Statistical calibration of computer simulations, *Reliab. Eng. Syst. Safety*, *91*, 1358–1363.

Conti, S., J. P. Gosling, J. E. Oakley, and A. O'Hagan (2009), Gaussian process emulation of dynamic computer codes, *Biometrika*, *96*, 663–676.

Cunge, J. (2003), Of data and models, *Hydroinformatics*, *5*, 75–96.

Dawson, R. J., and J. W. Hall (2006), Adaptive importance sampling for risk analysis of complex infrastructure systems, *Proc. R. Soc. A.*, *462*, 3343–3362.

Dawson, R. J., J. W. Hall, P. B. Sayers, P. D. Bates, and C. Rosu (2005), Sampling-based flood risk analysis for fluvial dike systems, *Stochastic Environ. Res. Risk Anal.*, *19*, 388–402.

Development Core Team (2005), *R: A Language and Environment for Statistical Computing*, 409 pp., R Found. for Stat. Comput., Vienna, Austria.

Duan, Q., N. K. Ajami, X. Gao, and S. Sorooshian (2007), Multi-model ensemble hydrologic prediction using Bayesian model averaging, *Adv. Water Resour.*, *30*, 1371–1386, doi:10.1016/j.advwatres.2006.11.014.

Engman, E. T. (1986), Roughness coefficients for routing surface runoff, *J. Irrig. Drain, Div.*, *112*, 39–53.

Gamerman, D. (1997), *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, 245 pp., Chapman and Hall, London.

Gelman, A. (1996), Inference and monitoring convergence, in *Markov Chain Monte Carlo in Practice*, edited by W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, pp. 131–144, Chapman and Hall, London.

Goldstein, M., and J. C. Rougier (2004), Probabilistic formulations for transferring inferences from mathematical models to physical systems, *SIAM J. Sci. Comput.*, *26*, 467–487.

Goldstein, M., and J. C. Rougier (2009), Reified Bayesian modelling and inference for physical systems, *J. Stat. Plann. Inference*, *139*, 1221–1239.

Hall, J. W. (2003), Handling uncertainty in the hydroinformatic process, *Hydroinformatics*, *5*, 215–232.

Hall, J. W., and D. Solomatine (2008), A framework for uncertainty analysis in flood risk management decisions, *J. River Basin Manage.*, *6*, 85–98.

Hall, J. W., S. Tarantola, P. D. Bates, and M. S. Horritt (2005), Distributed sensitivity analysis of flood inundation model calibration, *J. Hydraul. Eng.*, *131*, 117–126.

Hankin, R. (2005), Introducing BACCO, an R bundle for Bayesian analysis of computer code output, *J. Stat. Software*, *14*, 21.

Higdon, D., M. C. Kennedy, J. Cavendish, J. Cafeo, and R. D. Ryne (2004), Combining field data and computer simulations for calibration and prediction, *SIAM J. Sci. Comput.*, *26*, 448–466.

Horritt, M. S., and P. D. Bates (2001), Predicting floodplain inundation: Raster-based modelling versus the finite element approach, *Hydrol. Processes*, *15*, 825–842.

Horritt, M. S., and P. D. Bates (2002), Evaluation of 1-D and 2-D numerical models for predicting river flood inundation, *J. Hydrol.*, *268*, 87–99.

Kavetski, D., G. Kuczera, and S. W. Franks (2006a), Calibration of conceptual hydrological models revisited: 1. Overcoming numerical artifacts *J. Hydrology*, *320*, 173–186, doi:10/1016/j.hydrol.2005.07.013.

Kavetski, D., G. Kuczera, and S. W. Franks (2006b), Calibration of conceptual hydrological models revisted: 2. Improving optimisation and analysis *J. Hydrology*, *320*, 187–201, doi:10.1016/j.hydrol.2005.07.013.

Kennedy, M. C., and A. O'Hagan (2001a), Bayesian calibration of computer models (with discussion), *J. R. Stat. Soc. Ser. B*, *63*, 425–464.

Kennedy, M. C., and A. O'Hagan (2001b), *Supplementary Details on Bayesian Calibration of Computer Models*, Univ. of Sheffield, Sheffield, U. K. (available at http://www.shef.ac.uk/∼st1ao/ps/calsup.ps)

Krzysztofowicz, R. (2001), The case for probabilistic forecasting in hydrology, *J. Hydrol.*, *249*, 2–9, doi:10.1016/S0022-1694(01)00420-6.

Krzysztofowicz, R. (1999), Bayesian theory of probabilistic forecasting via deterministic hydrologic model, *Water Resour. Res.*, *35*(9), 2739–2750.

Liu, F., and M. West (2009), A dynamic modelling strategy for Bayesian computer model emulation, *J. Bayesian Anal.*, *4*(2), 393–412, doi:10.1214/09-BA415.

Mantovan, P., and E. Todini (2006), Hydrological forecasting uncertainty assessment: Incoherence of the GLUE methodology, *J. Hydrol.*, *330*, 368–381.

Mardia, K., J. Kent, and J. Bibby (1979), *Multivariate analysis*, 518 pp., Harcourt Brace, London.

Montanari, A. (2005), Large sample behaviors of the generalized likelihood uncertainty estimation (GLUE) in assessing the uncertainty of rainfall-runoff simulations, *Water Resour. Res.*, *41*, W08406, doi:10.1029/2004WR003826.

Montanari, A., and A. Brath (2004), A stochastic approach for assessing the uncertainty of rainfall-runoff simulations, *Water Resour. Res.*, *40*, W01106, doi:10.1029/2003WR002540.

Moradkhani, H., K.-L. Hsu, H. Gupta, and S. Sorooshian (2005), Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter, *Water Resour. Res.*, *41*, W05012, doi:10.1029/2004WR003604.

Oakley, J. (2002), Eliciting Gaussian process priors for complex computer codes, *Statistician*, *51*, 81–97.

Oakley, J., and A. O'Hagan (2002), Bayesian inference for the uncertainty distribution of computer model outputs, *Biometrika*, *89*, 769–784.

Oberstadler, R., H. Hoensch, and D. Huth (1997), Assessment of the mapping capabilities of ERS-1 SAR data for flood mapping: a case study in Germany, *Hydrol. Processes*, *10*, 1415–25.

O'Hagan, A., and J. J. Forster (2004), *Bayesian Inference, Kendall's Advanced Theory of Statistics*, 352 pp., Arnold, London.

Raftery, A. E., and S. M. Lewis (1996), Implementing MCMC, in *Markov Chain Monte Carlo in Practice*, edited by W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, pp. 115–130, Chapman and Hall, London.

Renard, B., D. Kavetski, G. Kuczera, M. A. Thyer, and S. W. Franks (2010), Understanding predictive uncertainty in hydrologic modeling: The challenge of identifying input and structural errors. *Water Resour. Res.*, *46*, W05521, doi:10.1029/2009WR008328.

Schumann, G., R. Hostache, C. Puech, L. Hoffmann, P. Matgen, F. Pappenberger, and L. Pfister (2007), High-resolution 3-D flood information from radar imagery for flood hazard management, *IEEE Trans. Geosci. Remote Sens.*, *45*(Part 1), 1715–1725.

Shrestha, D. L., and D. P. Solomatine (2006), Machine learning approaches for estimation of prediction interval for the model output, *Neural Networks*, *19*, 225–235.

Thyer, M., B. Renard, D. Kavetski, G. Kuczera, S. W. Franks, and S. Srikanthan (2009), Critical evaluation of parameter consistency and predictive uncertainty in hydrological modeling: A case study using Bayesian total error analysis, *Water Resour. Res.*, *45*, W00B14, doi:10.1029/2008WR006825.

Todini, E. (2004), Role and treatment of uncertainty in real-time flood forecasting, *Hydrol. Processes*, *18*, 2743–2746.

Todini, E. (2007), Hydrological catchment modelling: Past, present and future, *Hydrol. Earth Syst. Sci.*, *11*, 468–482.

Todini, E. (2008), Predictive uncertainty in flood forecasting models, *J. River Basin Manage.*, *6*, 123–137.

Vanmarcke, E. (1983), *Random Fields: Analysis and Synthesis*, 396 pp., MIT Press, Cambridge, Mass.

Vrugt, J. A., C. G. H. Diks, H. V. Gupta, W. Bouten, and J. M. Verstraten (2005), Improved treatment of uncertainty in hydrologic modeling: Combining the strengths of global optimization and data assimilation, *Water Resour. Res.*, *41*, W01017, doi:10.1029/2004WR003059.

Yen, B. C. (1992), *Channel Flow Resistance: Centennial of Manning's Formula*, 534 pp., Water Res. Pub., Highlands Ranch, Denver, Colo.

J. W. Hall, Environmental Change Institute, University of Oxford, Oxford OX1 2QY, UK. (jim.hall@eci.ox.ac.uk)

R. K. S. Hankin, Department of Land Economy, 19 Silver St., Cambridge CB3 9EP, UK.

L. J. Manning, School of Civil Engineering and Geosciences, Cassie Building, Newcastle upon Tyne NE1 7RU, UK.