

Impact of land use on water resources via a Gaussian process emulator with dimension reduction

Nathan E. Owen and Lorena Liuzzo

ABSTRACT

The replacement of models by emulators is becoming a frequent approach in environmental science due to the reduction of computational time, and different approaches exist in the water resources modelling literature. In this work, an emulator to mimic a hydrological model at catchment scale is proposed, taking into account the effect of land use on the hydrological processes involved in water balance. The proposed approach is novel for its combination of techniques. The dimension of the temporal model output is reduced via principal component analysis, and this reduced output is approximated using Gaussian process emulators built on a conditioned Latin hypercube design to reflect constrained land use inputs. Uncertainty from both the model approximation and the dimension reduction is propagated back to the space of the original output. The emulator has been applied to simulate river flow in a rural river basin located in south west England, the Frome at East Stoke Total, but the methodology is general. Results showed that the use of the emulator for water resources assessment at catchment scale is an effective approach, providing accurate estimates of the model output as a function of land use inputs, for a reduction of the computational burden.

Key words | dimension reduction, mixture experiments, runoff modelling, soil moisture balance, statistical emulator, uncertainty quantification

Nathan E. Owen
Lorena Liuzzo (corresponding author)
Land, Environment, Economics and Policy Institute
(LEEP),
University of Exeter,
XFI Building, Exeter,
UK
E-mail: lorenaliuzzo@gmail.com

INTRODUCTION

Environmental modelling, particularly for the simulation of geological and climatic processes, usually requires the use of computationally expensive and long runtime codes. This is an issue for analyses that require a large number of model simulations, such as a sensitivity analysis or model calibration, and such analyses can become prohibitive or even impossible using the original model. Recently, the development of new methodologies has been explored, in order to approximate complex environmental models with simpler and more computationally efficient models (e.g. [Castelletti et al. 2009](#); [Fu et al. 2010](#); [Lima et al. 2018](#)). One of the most frequently used approaches is emulation or metamodelling. According to the definition given by [Castelletti et al. \(2012\)](#), an emulator can be considered as a lower-order approximation of a complex, process-based model,

designed to be used as a surrogate in order to overcome the large computational burden. In general, the purpose of an emulator is to replicate the functions of one system with a different system that attempts to reproduce its behaviour ([Abrahart & See 2007](#); [Stanfill et al. 2015](#)). A variety of different emulators have been developed and tested in a wide range of applications, such as river basin and groundwater management ([Solomatine & Ostfeld 2008](#); [Mirfenderesgi & Mousavi 2015](#); [Roy et al. 2016](#); [Christelis et al. 2017](#)), water quality modelling ([Castelletti et al. 2010](#)), computational fluid dynamics ([Moonen & Allegrini 2015](#)), fire emission modelling ([Katurji et al. 2015](#)), and ecological modelling ([Leeds et al. 2013](#)).

Hydrological modelling is a field which benefits greatly from the use of emulators, not only for facilitating common

tasks such as sensitivity analysis or calibration, but also for other applications, such as the investigation of effects of climate or land use change on water resources. Indeed, this kind of analysis requires a great number of model runs in order to assess the uncertainty related to predictions in potential future scenarios. Metamodelling approaches have been widely applied in various hydrological modelling problems, as reported by [Razavi *et al.* \(2012\)](#) who reviewed and analyzed a number of studies on emulators in the water resources modelling field. One of the first applications of emulation in hydrology can be attributed to [Romanowicz *et al.* \(1996\)](#), who developed a distributed flow routing model emulator to explore the accuracy in flood prediction. In order to predict water level in the Severn River basin (UK), [Young *et al.* \(2009\)](#) presented and applied a nonlinear Dynamic Model Emulator (DME) to reproduce the output of the HEC-RAS model ([Brunner 2001](#)). [Fraser *et al.* \(2013\)](#) proposed the application of a metamodelling approach to improve the performance of a physically-based model in the assessment of the implications of land use changes on peak flows at the catchment scale. [Romanowicz & Osuch \(2015\)](#) developed an emulator based on the MIKE11 model ([DHI 1997](#)), obtaining reliable flow predictions for a basin of the river Vistula in Poland. More recently, [Machac *et al.* \(2016\)](#) and [Carbajal *et al.* \(2017\)](#) investigated the application of emulation in urban drainage modelling. [Machac *et al.* \(2016\)](#) proposed an emulator to simplify the application of the Storm Water Management Model (SWMM) to the urban drainage system of the city of Adliswil (Switzerland). [Carbajal *et al.* \(2017\)](#) compared the performance of two different kinds of emulators, mechanistic and data-driven, to mimic the results obtained by the application of an urban drainage model of two small catchments in Switzerland, finding that data-driven emulation outperforms mechanistic emulation where the mechanistic prior is linear.

In this work, a statistical emulator has been developed to mimic and replace a hydrological model based on a soil moisture balance ([Rushton *et al.* 2006](#)). This model is quite simplistic in nature and has mainly been chosen to demonstrate the emulator methodology. The proposed approach is unique as it combines a range of techniques to develop an efficient emulator, namely: the use of principal component analysis ([Jolliffe 2002](#)) to reduce the high dimensional

nature of the model output; the use of models from the field of mixture experiments ([Cornell 2011](#)) which are appropriate for the constrained land use model inputs; a novel design strategy called conditioned Latin hypercube sampling (CLHS) which combines traditional mixture design with Latin hypercubes; and the use of Gaussian process emulation ([Kennedy & O'Hagan 2001](#); [Rasmussen & Williams 2006](#)) for accurate prediction of the model output as well as uncertainty quantification. The uncertainty from both the emulation of the model and the dimension reduction is also explicitly propagated back to the space of the original outputs. The proposed methodology has been applied to simulate river flow in a rural river basin in south west England, the Frome at East Stoke Total. This work focuses on a hydrological application, but the statistical emulator presented here is a general technique which could be used for a range of environmental models. The approach presented here is similar to the data-driven emulator of [Carbajal *et al.* \(2017\)](#), with the addition of the novel design strategy for the constrained inputs, explicit calculation of the uncertainty propagation, and a different test model.

The paper is organized as follows. In the next section, the area of study and datasets are presented and described. The proposed methodology is explained in the following section. Results are presented and discussed are then discussed and in the next section, applied methodologies and obtained results are briefly summarized.

STUDY AREA AND DATA

The study area is the Frome at East Stoke Total river basin, located in south west England. The catchment area is approximately 415 km². In this area, land cover includes pasture (90%), arable (8.7%), woodlands (1%), and urban areas (0.3%). The prevalent crop in the arable areas is winter wheat. [Figure 1](#) shows the digital elevation model of the basin, the outlet location and the spatial distribution of land cover.

The geology is characterized by Chalk, Upper Greensand and Gault, Lias and Oolites in headwaters area, while sand, gravels and clays are frequent in the lower part of the catchment. The predominant soil type in the

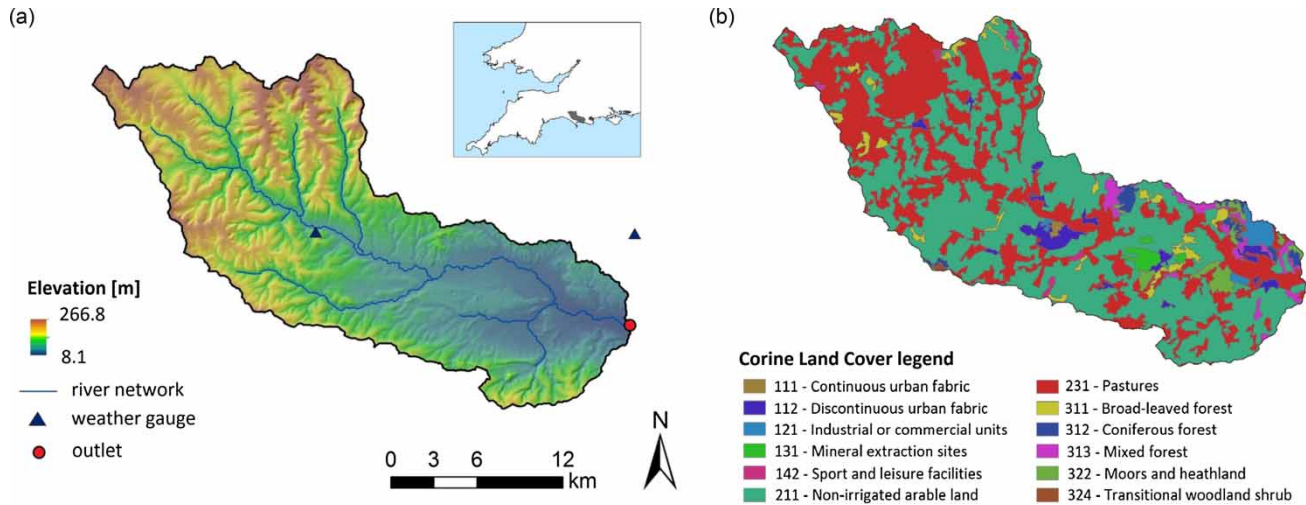


Figure 1 | (a) Digital elevation model of the river basin; (b) land cover map (Feranec *et al.* 2016).

river basin is Rendzina soil, covering 43.2% of the area, whereas the remaining soils are Eutric Cambisols (27.4%), Orthic Podzols (20.8%) and Eutric Gleysols (8.6%).

According to the Köppen classification, the climate of south west England is classified as oceanic, characterized by cool winters and warm summers. The average annual temperature is approximately 10 °C. Summer is the warmest season with an average annual temperature of 15–16 °C. Precipitations are distributed during the whole year. Average annual rainfall ranges between 1,000 and 1,100 mm/year.

The average annual flow at the outlet of the Frome river basin is approximately 6.7 m³/s. In the catchment, the natural flow regime is not affected by upstream water abstractions, nevertheless, a considerable amount of water is abstracted from the chalk area by bore holes and wells.

Daily series of precipitation, temperature, solar radiation, relative humidity and wind speed data have been obtained from a dataset based on the global data of National Centers for Environmental Prediction (NCEP) Climate Forecast System Reanalysis (CFSR), available at the website <http://globalweather.tamu.edu/>. In the website, the selected weather gauges (all located in the same site with coordinates 50°44'13.19"N and 2°29'59.98"W) are identified with the following IDs: p507-25 (rainfall in mm), t507-25 (maximum and minimum temperatures in °C), r507-25 (relative humidity as a fraction), s507-25 (solar radiation in MJ/m²) and w507-25 (wind speed in m/s). Daily river flow data

recorded at the East Stoke Total station (9.2 mAOD) have been provided by the National River Flow Archive of UK. In this dataset, the river flow gauge is identified with the ID 44001 and data can be downloaded at the link <https://nrfa.ceh.ac.uk/data/station/meanflow/44001>. Land cover information in the area of study have been obtained from Feranec *et al.* (2016) (downloadable at <https://land.copernicus.eu/pan-european/corine-land-cover/clc-2012?tab=download>), while soil information has been extracted from the FAO/UNESCO Soil Map of the World (downloadable at www.fao.org/geonetwork/srv/en/metadata.show%3Fid=14116).

METHODOLOGY

Runoff modelling

The applied model has been developed by Rushton *et al.* (2006). Based on soil moisture conditions, that are a function of catchment characteristics, such as the crop types in arable areas and the related growth stages, the model is able to simulate runoff and potential groundwater recharge.

The processes included in the soil moisture balance can be summarized in the following steps (Figure 2): precipitation reaches the ground surface and, based on rainfall intensity and soil moisture conditions, a fraction of it can generate runoff, whereas another part infiltrates the soil

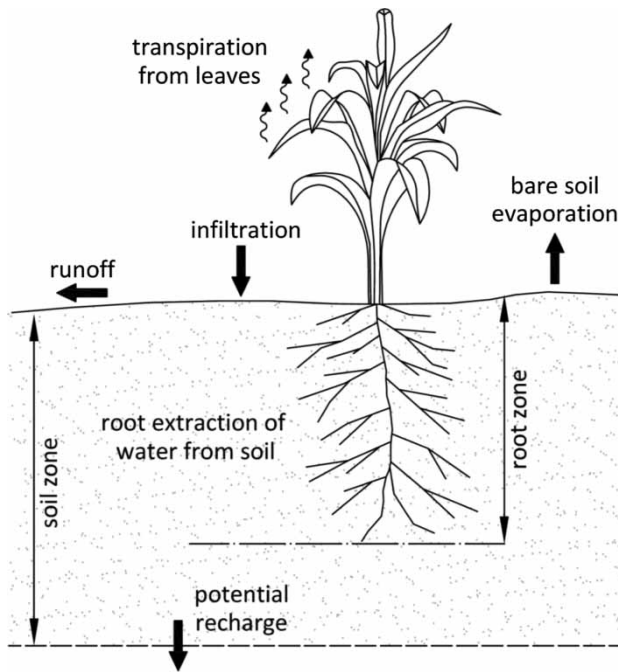


Figure 2 | Processes involved by the runoff model in the soil moisture balance.

zone. The amount of infiltrated water becomes available to supply potential evapotranspiration or to be stored as groundwater. Every time the potential evapotranspiration demand has been met, any further infiltration of water will increase the amount of water stored in the soil. This process will continue until the stored water reaches the field capacity, that is the point at which the soil becomes free draining and groundwater recharge occurs. During dry periods, the soil dries because of water uptake by the crop. The deficit of soil moisture represents the amount of water needed to bring the soil back to the field capacity. Whenever enough water is available, actual evapotranspiration will occur at the potential rate.

To identify when the soil reaches the field capacity throughout a year, soil moisture conditions are simulated on a daily basis. Soil moisture distribution in the soil is related to the permanent wilting point θ_{WP} , that is, the soil moisture content below which the plant roots are not able to extract water, and the field capacity θ_{FC} . Rainfall is the main input for soil moisture balance, while interception and runoff reduce the actual infiltration to the soil zone. Runoff can be assessed by means of coefficients that define the fraction of rainfall which generates runoff.

These coefficients are a function of daily rainfall and soil moisture deficit and are defined in order to reflect the small runoff in summer (higher soil moisture deficit) and the larger runoff in winter (lower soil moisture). The reference crop potential evapotranspiration ET_0 is estimated using the FAO version of the Penman–Monteith equation (Allen *et al.* 1998). The amount of water required by each crop depends on different factors, such as the date of sowing and harvest, the growth of crop, etc. The potential evapotranspiration of the crop ET_C is a function of ET_0 and is calculated as follows:

$$ET_C = K_S \cdot K_C \cdot ET_0 \quad (1)$$

where K_S is the stress coefficient and K_C is the crop coefficient (Allen *et al.* 1998). K_S is calculated as a function of root depth, the moisture holding properties of the soil and the soil moisture deficit, whereas K_C depends on the growth stage of the crop, assuming that the highest values are ascribed to the development and the mid stage, when the crop matures. Evaporation from bare soil occurs from crop harvest up to the complete coverage of ground by the crop.

The total water available to the plants (TAW) depends on the depth of the crop roots Z_r and the difference between the water content at the field capacity and the wilting point. TAW is calculated as follows:

$$TAW = 1000(\theta_{FC} - \theta_{WP})Z_r \quad (2)$$

A further variable, the readily available water (RAW), is introduced in the model in order to represent the condition in which the crop can obtain all the required water from the soil:

$$RAW = p \cdot TAW \quad (3)$$

where p is a depletion factor ranging from 0.2 and 0.7. Every time the soil moisture deficit exceeds RAW , the crop is under stress conditions, then transpiration will occur at a reduced rate, while when RAW is higher than the soil moisture deficit, the crop will transpire at the potential rate. Groundwater recharge will occur on days when the soil

moisture deficit becomes zero and the soil becomes free draining. Groundwater recharge is equal to the amount of water exceeding that required by the soil to reach the field capacity (see Appendix, available with the online version of this paper). All stages of the soil moisture balance calculations and equations are fully described by Rushton *et al.* (2006) and Ireson & Butler (2013).

The model application requires the calibration of six parameters, specifically, θ_{FC} , θ_{WP} , p , Zr , the bypass fraction BF and the factor for near surface storage $FRACSTOR$. Model outputs are very sensitive to changes of the BF parameter (Ireson & Butler 2013), representing the proportion of rainfall which becomes bypass flow, that is the amount of water passing directly to the aquifer through the fractures. Bypass flow is responsible for groundwater recharge during summer months, when soil moisture deficits prevent drainage from the soil. $FRACSTOR$ is an empirical factor used to define the proportion of the increase in moisture content that becomes near surface soil storage.

The model previously described has been calibrated and validated. In order to identify the set of model parameters that give a good fit between observed and simulated discharges, a Monte Carlo procedure has been applied, consisting of a random generation of parameter values in fixed ranges of variation. Therefore, each model simulation for calibration has been characterized by a different combination of randomly generated parameters. The achievement of the best set of model parameters required a great number of simulations (10,000), since several combinations of parameters need to be tested.

The fact that the Rushton *et al.* (2006) model is fast enough to enable a Monte Carlo calibration approach contradicts the need for emulation somewhat. However, the focus of this paper is to outline the methodology of the emulator and its implementation. The aforementioned model is used for demonstration purposes and has been pre-calibrated in order to avoid a biased poor performance of the emulator. Indeed, statistical emulators are commonly employed in the calibration stage itself, but this requires a slightly different approach to the focus here. A number of calibration approaches have been proposed in the emulation literature, including Bayesian calibration (Kennedy & O'Hagan 2001; Wilkinson 2011) and history matching (Craig *et al.* 1996; Williamson *et al.* 2015).

The model performance has been evaluated through the Nash and Sutcliffe coefficient (Nash & Sutcliffe 1970), an efficiency index commonly used in hydrological modelling:

$$NS = 1 - \frac{\sum_{t=1}^T (y_{obs,t} - y_{sim,t})^2}{\sum_{t=1}^T (y_{obs,t} - \bar{y}_{obs})^2} \quad (4)$$

where $y_{obs,t}$ and $y_{sim,t}$ are the observed and simulated river flow at time t , and \bar{y}_{obs} is the average of the observed series. NS can range from $-\infty$ to 1. An efficiency equal to 1 is reached when a perfect match between observed and simulated values occurs.

Model set up and experimental design

With the calibration complete, the aim is to study the relationship between the remaining four land use input variables and the river flow output. The land use variables are defined as shares in the basin and will be denoted as x_1 (woodland), x_2 (arable), x_3 (pasture) and x_4 (urban), or using the vector $\mathbf{x} = (x_1, x_2, x_3, x_4)$. The model output is the daily river flow at T time points, denoted as the vector $\mathbf{y} = (y_1, \dots, y_T)$. The Rushton *et al.* (2006) model will be denoted functionally as $\mathbf{y} = \eta(\mathbf{x})$. To reduce the computational burden, the model will be replaced by a fast-to-run statistical emulator, denoted $\mathbf{y} \approx \hat{\eta}(\mathbf{x})$. The methodology for building the statistical emulator is presented below under 'Statistical emulation'. An analysis of the difference in computational time between the model and emulator is given below under 'Comparing the computational time of model and emulator'. Daily river flow output of the Rushton *et al.* (2006) model for five settings of the four land use inputs are shown in Figure 3.

The statistical emulator must be constructed using n runs of the model in an experimental design, denoted $\mathbf{X} = (\mathbf{x}^1, \dots, \mathbf{x}^n)'$. The experimental design literature for statistical emulation is vast, but in general it is good practice to run the model at a range of input settings which are space-filling in the region of interest, and Latin hypercubes (McKay *et al.* 1979) are a popular approach. The number of model runs, n , depends on the computational budget, but Loepky *et al.* (2009) suggest the rule $n = 10d$, where d denotes the input dimension ($d = 4$ here). The land use inputs for the Rushton *et al.* (2006) model are non-standard

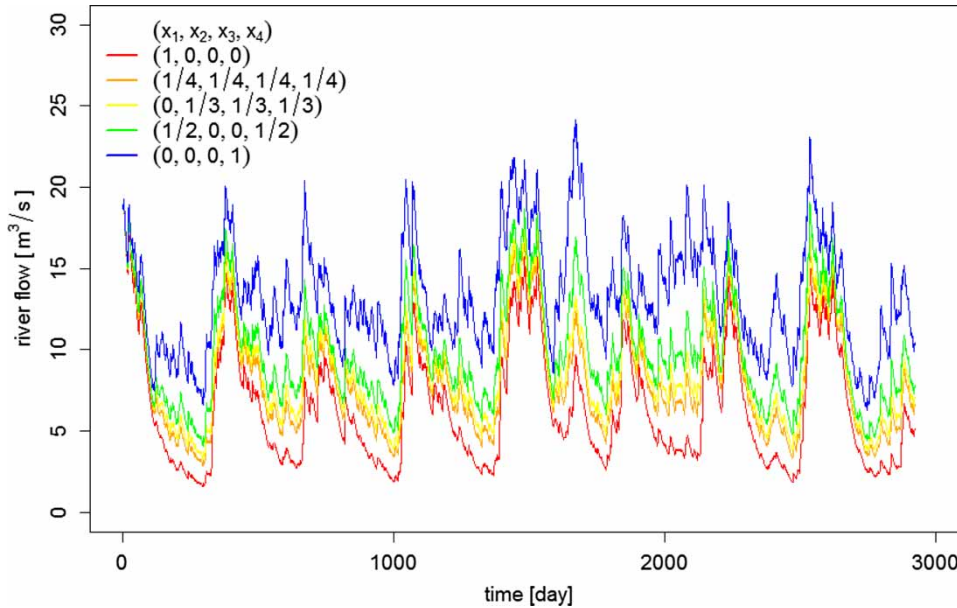


Figure 3 | Daily river flow output for five settings of the four land use inputs x_1 (woodland), x_2 (arable), x_3 (pasture) and x_4 (urban).

in that they are shares constrained to sum to one, i.e. $\sum_{j=1}^4 x_j = 1$. This is the focus of the field of mixture experiments (Cornell 2011) and several mixture designs have been suggested. However, the majority of mixture designs, e.g. simplex centroid designs and simplex lattice designs, are similar to factorial or gridded designs, which are known to be substandard for building (Gaussian process) statistical emulators (Urban & Fricker 2010). As far as the authors are aware, equivalents of Latin hypercube designs for mixture experiments do not exist in the literature. Here, a novel strategy for mixture design for emulators is proposed which uses the CLHS algorithm (Minasny & McBratney 2006). The algorithm is implemented in the *clhs* package (Roudier 2011) in the R programming language (R Core Team 2017). With the design strategy decided it remains to choose the number of training runs (design size). In this work, a CLHS design of size $n = 40$ is chosen, following the rule $n = 10d$ suggested by Loepky *et al.* (2009), from 1000 candidate mixture points. The experimental design is shown in Figure 4. The $n = 10d$ rule of thumb is satisfactory in many scenarios, and provides accurate emulator predictions here (see below under ‘Statistical emulation’). However, in cases where there are a large number of inputs or the model is too expensive, it may not

be possible to achieve within the computational limitations. A solution in this case would be to reduce the input dimension, for example using active subspaces (Constantine 2015), or to implement smaller experiments over multiple iterations or waves (Williamson *et al.* 2015).

In summary, the experimental design \mathbf{X} is a CLHS mixture design of size $n = 40$. The output of the Rushton *et al.* (2006) model at the experimental design is the matrix $\mathbf{Y} = (\mathbf{y}^1, \dots, \mathbf{y}^n)'$, obtained by evaluating $\mathbf{y}^i = \eta(\mathbf{x}^i)$ for $i = 1, \dots, n$. Since the river flow model output is strictly positive, the logarithm of the model output, $\mathbf{Y}_l = \log(\mathbf{Y})$, is used for subsequent statistical analysis.

Dimension reduction

The majority of statistical emulation methods are appropriate for scalar model outputs (Kennedy & O’Hagan 2001). To adhere to this and use standard emulation techniques, the dimension of the temporal river flow model output is reduced. In this work, principal component analysis is employed via the singular value decomposition (SVD) (Jolliffe 2002). Several other decompositions are possible, for example nonnegative matrix factorization (Carbajal *et al.* 2017). The dimension reduction procedure used here is as follows:

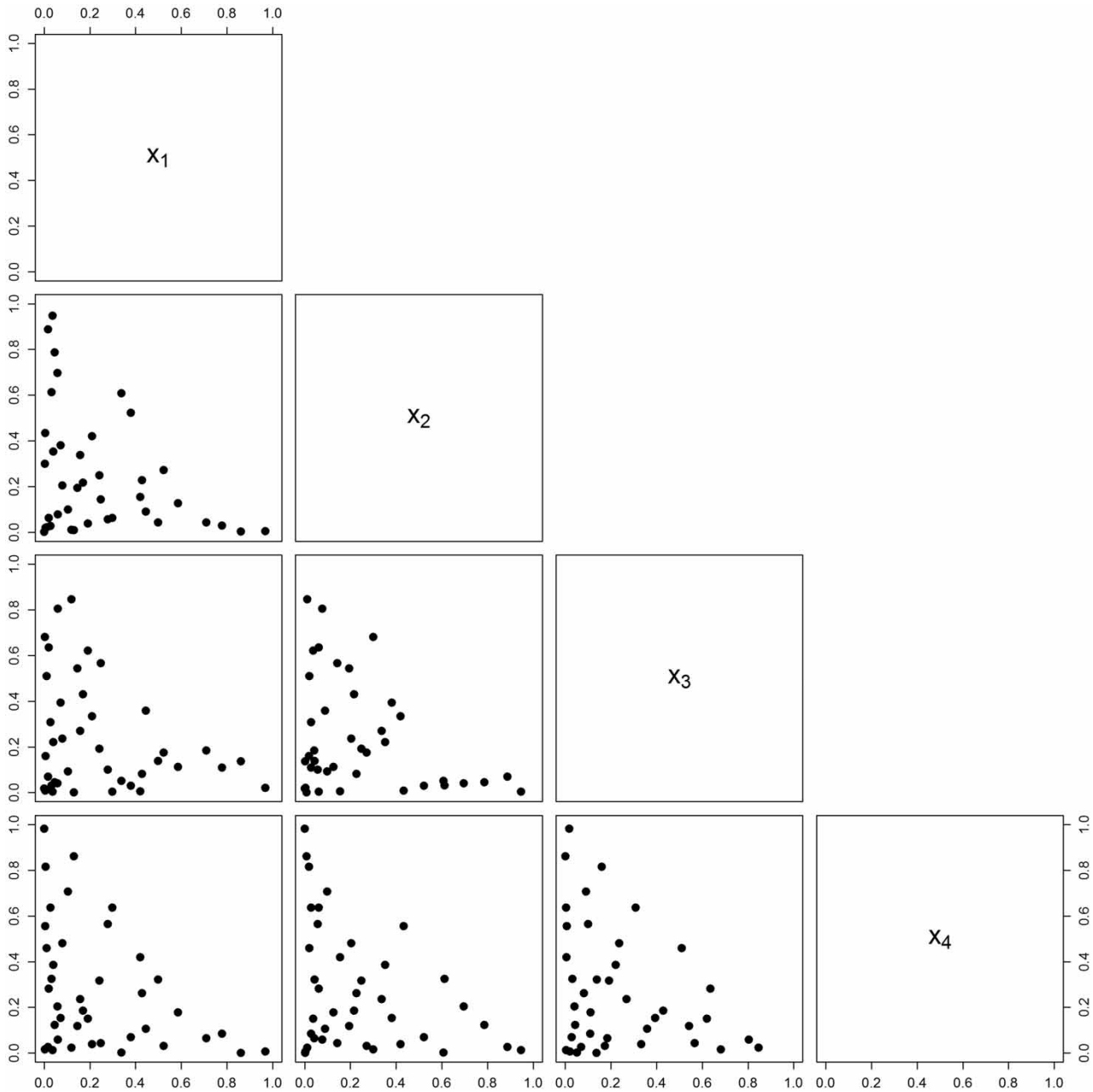


Figure 4 | Z Scatterplot matrix of the $n = 40$ experimental design points in the four land use inputs x_1 (woodland), x_2 (arable), x_3 (pasture) and x_4 (urban), used to build statistical emulators.

1. Subtract the column means \mathbf{m} from \mathbf{Y}_l to obtain $\tilde{\mathbf{Y}} = \mathbf{Y}_l - \mathbf{m}$.
2. Using SVD (or equivalently an eigendecomposition) obtain a column matrix of eigenvectors, \mathbf{V} , and a diagonal matrix of eigenvalues, $\mathbf{\Lambda}$. The eigenvalues satisfy $\lambda_1 \geq \dots \geq \lambda_T$.
3. Select the first K eigenvectors and eigenvalues to explain a proportion P (e.g. 0.9, 0.95, 0.99) of variability in the data, by solving for K : $\sum_{k=1}^K \lambda_k / \sum_{k=1}^T \lambda_k \geq P$. Denote the first K eigenvectors and eigenvalues using the matrices \mathbf{V}_K and $\mathbf{\Lambda}_K$ respectively. Denote the remaining

eigenvectors and eigenvalues, $k = K + 1, \dots, T$, using the matrices \mathbf{V}_{-K} and $\mathbf{\Lambda}_{-K}$ respectively.

4. Project the data by computing $\tilde{\mathbf{Y}}_K = \tilde{\mathbf{Y}}\mathbf{V}_K$. The columns of $\tilde{\mathbf{Y}}_K$ are the principal components, denoted $\tilde{y}_1, \dots, \tilde{y}_K$.
5. Transform back to the original dimension of the data by computing $\mathbf{Y}_K = \tilde{\mathbf{Y}}_K\mathbf{V}'_K + \mathbf{m}$. Since a logarithmic transformation was used, an exponential transformation will also be needed here.

By using principal component analysis on \mathbf{Y}_l it was found that $K = 2$ eigenvectors could explain almost all of the variability in the data ($P \approx 1$). Usual practice is to select the number of eigenvectors to explain a high proportion of variability in the data (e.g. 0.9, 0.95, 0.99). Since almost all the variability could be explained by only two eigenvectors in our example it made sense to go beyond this rule. This is very uncommon and is probably due to the model output being similar across model runs, as we can see in Figure 3. As the land use input is varied the flow output is similar, save for a shift effect. Despite explaining a high proportion of variability, the principal component approach does not guarantee accurate reconstruction of the original data, especially if there is non-linearity present. However, it is shown that the reconstruction is accurate below under 'Results and discussion'.

Statistical emulation

Recall that the Rushton *et al.* (2006) model, $\mathbf{y} = \eta(\mathbf{x})$, is to be replaced with an emulator, $\mathbf{y} \approx \hat{\eta}(\mathbf{x})$. Using principal components the model output \mathbf{y} has been approximated by the two principal components, \tilde{y}_1 and \tilde{y}_2 . To construct a full emulator for the Rushton *et al.* (2006) model, functional relationships between the land use inputs and the principal components, denoted $\tilde{y}_1 = \eta_1(\mathbf{x})$ and $\tilde{y}_2 = \eta_2(\mathbf{x})$, need to be replaced with independent emulators, namely $\tilde{y}_1 \approx \hat{\eta}_1(\mathbf{x})$ and $\tilde{y}_2 \approx \hat{\eta}_2(\mathbf{x})$.

In this work, a popular technique called Gaussian process emulation will be used (Kennedy & O'Hagan 2001). A short overview of Gaussian process emulation will be given here; for more information refer to Rasmussen & Williams (2006) or the tutorial given by O'Hagan (2006). A Gaussian process is an extension of the

univariate and multivariate Gaussian probability distributions to describe the properties of functions, rather than scalars or vectors. It is completely defined by its mean and covariance function, similar to the mean and variance parameters of a Gaussian probability distribution. Used as a statistical emulator for a model, the Gaussian process is usually employed in a Bayesian framework (e.g. Kennedy & O'Hagan 2001; O'Hagan 2006). The initial step is to assume a prior distribution for the model output (the two principal components here), and that this can be modelled using a Gaussian process. In this work, the following prior distribution will be used for the k th principal component:

$$\eta_k(\mathbf{x}) \mid \boldsymbol{\beta}_k, \sigma_k^2, \boldsymbol{\delta}_k \sim GP(M_k(\mathbf{x}, \boldsymbol{\beta}_k), V_k(\mathbf{x}, \mathbf{x}'; \sigma_k^2, \boldsymbol{\delta}_k)) \quad (5)$$

where $M_k(\cdot)$ and $V_k(\cdot, \cdot)$ denote the mean and covariance functions respectively, and $\boldsymbol{\beta}_k$, σ_k^2 and $\boldsymbol{\delta}_k$ are hyperparameters to be estimated. The mean function is used to describe any known global behaviour of the model output as a function of its inputs, and commonly a regression model is used. Recall that for the Rushton *et al.* (2006) model the land use inputs are shares constrained to sum to one. It is therefore appropriate to use a regression model from the field of mixture experiments, the most popular class of which are Scheffé models (Scheffé 1958; Cornell 2011). For simplicity, in this work the following linear Scheffé model is used:

$$M_k(\mathbf{x}, \boldsymbol{\beta}_k) = \beta_{1,k}x_1 + \beta_{2,k}x_2 + \beta_{3,k}x_3 + \beta_{4,k}x_4 \quad (6)$$

where the hyperparameters $\boldsymbol{\beta}_k = (\beta_{1,k}, \beta_{2,k}, \beta_{3,k}, \beta_{4,k})$ are regression coefficients to be estimated. This is to align with the fact that linear regression models are the standard choice for the mean function in Gaussian emulation (O'Hagan 2006). The covariance function is used to describe the smoothness properties of the model output as a function of its inputs, and there are several choices from the literature. The majority of these define the covariance in the output at two input settings, \mathbf{x} and \mathbf{x}' , in terms of the distance between the inputs, $\mathbf{x} - \mathbf{x}'$. In this work, the following squared exponential covariance function is used, which assumes a stationary and

separable Gaussian process (Rasmussen & Williams 2006):

$$V_k(\mathbf{x}, \mathbf{x}'; \sigma_k^2, \boldsymbol{\delta}_k) = \sigma_k^2 \exp\left(-\frac{(\mathbf{x} - \mathbf{x}')\Delta_k(\mathbf{x} - \mathbf{x}')'}{2}\right) \quad (7)$$

where Δ_k is a matrix with diagonal elements $1/\delta_{j,k}^2$, $j = 1, \dots, d$. The hyperparameters $\boldsymbol{\delta}_k = (\delta_{1,k}, \dots, \delta_{d,k})$ are known as correlation lengths, and determine the smoothness of the output in each input separately. Specifying one correlation length for each input dimension is a technique known as automatic relevance determination (ARD) (Rasmussen & Williams 2006). Satisfying $\delta_k > 0$, the larger the correlation length, the smoother the output is with respect to that input. The hyperparameter σ_k^2 is a variance term which determines the extent that the Gaussian process can deviate away from the mean function. The squared exponential covariance function results in a very smooth (infinitely differentiable) Gaussian process which will not be appropriate in all cases, but is found to be suitable for the principal component output featured here. However, the proposed approach will work with any other choice of covariance function, for example, the Matérn class of covariance functions. It is fundamental to correctly choose the covariance function. It is shown below under 'Statistical emulation' that the squared exponential covariance function with ARD gives the best value of the marginal log-likelihood when compared to Matérn covariance functions with ARD. To test whether the choice of the squared exponential covariance function with ARD is optimal, the value of the marginal log-likelihood is calculated and compared with values obtained by using Matérn covariance functions with ARD. Specifically, the squared exponential covariance function is compared with three Matérn covariance functions with ARD, where the regularity parameter is set to 1, 3 and 5. These are referred to as Matérn 1, Matérn 3 and Matérn 5 respectively. A larger value of the regularity parameter increases the smoothness of the resulting Gaussian process. More detail on the choice and role of the mean and covariance function and on the Matérn class of covariance functions can be found in Rasmussen & Williams (2006).

With the prior specification complete, the next step is to derive a posterior distribution conditional on runs of the model at an experimental design. The data required for

this calculation are vectors of the principal component outputs \tilde{y}_k , $k = 1, 2$, obtained at the experimental design \mathbf{X} . The posterior distribution for the k th principal component output, conditional on estimates of the hyperparameters $\hat{\boldsymbol{\beta}}_k$, $\hat{\sigma}_k^2$ and $\hat{\boldsymbol{\delta}}_k$, remains a Gaussian process and is as follows:

$$\eta_k(\mathbf{x}) | \tilde{\mathbf{y}}_k, \hat{\boldsymbol{\beta}}_k, \hat{\sigma}_k^2, \hat{\boldsymbol{\delta}}_k \sim GP(M_k^*(\mathbf{x}), V_k^*(\mathbf{x}, \mathbf{x}')) \quad (8)$$

where $M_k^*(\cdot)$ and $V_k^*(\cdot, \cdot)$ denote the posterior mean and covariance functions respectively. Full details of this calculation can be found in Haylock & O'Hagan (1996). This posterior distribution is known as the Gaussian process emulator, and provides a fast approximation to the model at any input setting. The posterior mean function $M_k^*(\mathbf{x})$ interpolates the model output \tilde{y}_k at the experimental design points in \mathbf{X} . The posterior uncertainty is captured by the variance $V_k^*(\mathbf{x}, \mathbf{x})$ and quantifies the uncertainty introduced by replacing the original model with an emulator. This uncertainty is zero at the experimental design points and grows as we move further away from them.

Reconstruction and propagation of uncertainty

With the statistical emulation of the principal components complete, it remains to reconstruct an emulator of the full Rushton *et al.* (2006) model and effectively propagate the uncertainty back to the space of the model output, the river flow time series. To do this, the procedure described in Wilkinson (2011) is followed. Assuming that the two principal components are independent, the predictive distribution of the combined statistical emulators at a new input setting \mathbf{x}^* is the following multivariate Gaussian distribution:

$$\eta_1(\mathbf{x}^*), \eta_2(\mathbf{x}^*) | \tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2 \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (9)$$

where $\boldsymbol{\mu} = (M_1^*(\mathbf{x}^*), M_2^*(\mathbf{x}^*))'$ is a mean vector and $\boldsymbol{\Sigma}$ is a covariance matrix with diagonal elements $V_k^*(\mathbf{x}^*, \mathbf{x}^*)$, $k = 1, 2$. This distribution can be propagated to the space of the (logarithm of the) Rushton *et al.* (2006) model output as follows (Wilkinson 2011):

$$\log(\eta(\mathbf{x}^*)) | \{\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2\} \sim N(\boldsymbol{\mu}\mathbf{V}'_K + \mathbf{m}, \mathbf{V}_K\boldsymbol{\Sigma}\mathbf{V}'_K + \mathbf{V}_{-K}\boldsymbol{\Lambda}_{-K}\mathbf{V}'_{-K}) \quad (10)$$

where the definitions of \mathbf{m} , \mathbf{V}_K , \mathbf{V}_{-K} and $\mathbf{\Lambda}_{-K}$ were given above under ‘Dimension reduction’. The term $\boldsymbol{\mu}\mathbf{V}'_K + \mathbf{m}$ is a propagation of the statistical emulator mean back to the space (of the logarithm) of the river flow output. The two variance terms $\mathbf{V}_K\boldsymbol{\Sigma}\mathbf{V}'_K$ and $\mathbf{V}_{-K}\mathbf{\Lambda}_{-K}\mathbf{V}'_{-K}$ quantify the uncertainty from replacing the model with an emulator, and reducing the dimension through K principal components, respectively. Like the mean term they themselves are propagated back to the space of the original model output. An exponential transformation of Equation (10) completes the reconstruction and propagation.

Validating the statistical emulator

It is important to validate a statistical emulator to see if it provides an accurate approximation to the model. In this work, leave-one-out cross validation is used to this end (Sammut & Webb 2017). The emulators are validated on the reduced dimensional (principal component) space and the reconstructed space of the original model output. A procedure for leave-one-out cross validation for the emulator $\hat{\eta}_k(\mathbf{x})$ in Equation (7) is as follows (for $i = 1, \dots, n$):

- Remove the i th point \mathbf{x}^i from the experimental design and the i th point $\tilde{\mathbf{y}}_k^i$ from the principal component output, denote the result as \mathbf{X}^{-i} and $\tilde{\mathbf{y}}_k^{-i}$ respectively.
- Construct a new Gaussian process emulator conditional on $\tilde{\mathbf{y}}_k^{-i}$; denote this $\hat{\eta}_k^{-i}(\mathbf{x})$. Instead of re-estimating the hyperparameters, simply fix them at the values estimated for the complete output $\tilde{\mathbf{y}}_k$. Denote the posterior mean and covariance functions of $\hat{\eta}_k^{-i}(\mathbf{x})$ as $M_k^{-i}(\mathbf{x})$ and $V_k^{-i}(\mathbf{x}, \mathbf{x}')$ respectively.
- Predict the principal component output at \mathbf{x}^i using the emulator. The predictive distribution is Gaussian with mean $M_k^{-i}(\mathbf{x}^i)$ and variance $V_k^{-i}(\mathbf{x}^i, \mathbf{x}^i)$.
- A $100(1 - \alpha)\%$ credible interval for the prediction can be constructed as $M_k^{-i}(\mathbf{x}^i) \pm t_{n-q-1; \alpha/2} \sqrt{V_k^{-i}(\mathbf{x}^i, \mathbf{x}^i)}$, where $t_{n-q-1; \alpha/2}$ denotes the $\alpha/2$ quantile of the Student's t -distribution with $n - q - 1$ degrees of freedom, and q is the number of $\boldsymbol{\beta}_k$ hyperparameters ($q = 4$ here).

Plotting leave-one-out cross validation emulator predictions against the corresponding principal component output provides a useful diagnostic validation tool. It is important to also include credible intervals to assess

whether the uncertainty of the emulator is well calibrated. In short, a well-calibrated emulator should have predictions close to the $y = x$ line with small uncertainty, or if not, the uncertainty should increase appropriately.

The emulator is also validated on the space of the original model output, after the reconstruction and propagation of uncertainty from ‘Reconstruction and propagation of uncertainty’ above has been applied. To do this, the simulated and emulated daily river flow time series are compared at the 40 leave-one-out cross validation points. The root mean square error (RMSE) and maximum absolute error (MAE) metrics are used to assess how well the emulator’s predictions match the model output:

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^T (y_t - \hat{y}_t)^2} \quad (11)$$

$$MAE = \max_{t=1, \dots, T} |y_t - \hat{y}_t| \quad (12)$$

where y_t and \hat{y}_t denote the simulated and emulated daily river flow at time t respectively.

RESULTS AND DISCUSSION

Runoff modelling

To calibrate the model, daily river flow data recorded over the 2003–2010 period has been used. Table 1 shows the set of parameters that provided the best model performance.

Table 1 | Parameters of calibrated model

Parameter	Description	Calibrated value	Measurement unit
θ_{FC}	Field capacity	0.201	m^3/m^5
θ_{WP}	Wilting point	0.096	m^3/m^3
p	Depletion factor for RAW calculation	0.495	–
Z_r	Root zone depth	0.539	m
BF	Bypass fraction	0.220	–
$FRACSTOR$	Factor for near surface storage	0.081	–

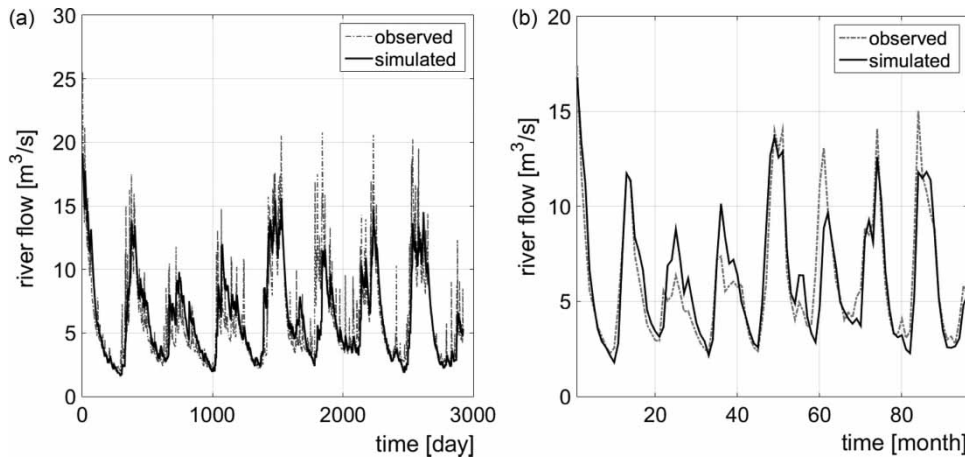


Figure 5 | Comparison between observed and simulated river flow with the calibrated model for the 2003–2010 period: (a) daily river flow, (b) monthly river flow.

At the daily scale, the calibrated model provided an *NS* coefficient equal to 0.71. The model performance improved at the monthly scale, reaching an *NS* value equal to 0.81. [Figure 5](#) shows the comparison between the observed river flow and the river flow simulated by the calibrated model in the 2003–2010 period at the daily ([Figure 5\(a\)](#)) and monthly scale ([Figure 5\(b\)](#)).

According to the previously described results, the model is able to reproduce the time pattern of the daily and average monthly river flow and the model performance can be considered acceptable ([Ritter & Muñoz-Carpena 2013](#)). Annual average observed river flow is equal to $6.2 \text{ m}^3/\text{s}$, while the simulated value is $6.3 \text{ m}^3/\text{s}$.

The set of best parameters were used to validate the model. Model validation has been performed using data measured from 2011 to 2015. Validation *NS* coefficients

were 0.64 and 0.70 for daily and monthly river flow respectively. These values of the efficiency index point out an acceptable model performance.

Statistical emulation

In this work, Gaussian process emulators are built for each principal component output using the DiceKriging package ([Roustant *et al.* 2012](#)) in *R*. The package implements maximum likelihood to estimate the hyperparameters $\hat{\beta}_k$, $\hat{\sigma}_k^2$ and $\hat{\delta}_k$ for the k th principal component output. Hyperparameter estimates for the emulators of the two principal components, $\tilde{y}_k \approx \hat{\eta}_k(\mathbf{x})$, $k = 1, 2$, are given in [Table 2](#).

The marginal log-likelihood of emulators $\hat{\eta}_k(\mathbf{x})$, $k = 1, 2$, with the four covariance function choices are given in [Table 3](#). It is clear that the choice of squared exponential covariance function provides the maximum value of the log-likelihood out of the four options, meaning that the principal component output is very smooth. This is also reflected in the

Table 2 | Hyperparameter estimates of the Gaussian process emulator $\hat{\eta}_k(\mathbf{x})$ in Equation (8) for the two principal component outputs \tilde{y}_k , $k = 1, 2$

Hyperparameter	$k = 1$	$k = 2$
$\hat{\beta}_{1,k}$	15.989	1.992
$\hat{\beta}_{2,k}$	15.237	-3.832
$\hat{\beta}_{3,k}$	14.673	1.967
$\hat{\beta}_{4,k}$	-40.707	-0.079
$\hat{\sigma}_k^2$	16.722	0.708
$\hat{\delta}_{1,k}$	1.938	1.938
$\hat{\delta}_{2,k}$	1.892	0.617
$\hat{\delta}_{3,k}$	1.691	1.691
$\hat{\delta}_{4,k}$	0.224	0.338

Table 3 | Marginal log-likelihood of the Gaussian process emulator $\hat{\eta}_k(\mathbf{x})$ in Equation (8) for the two principal component outputs \tilde{y}_k , $k = 1, 2$, with four choices of the covariance function

Covariance function	$k = 1$	$k = 2$
Squared exponential	13.434	71.017
Matérn 1	-58.845	17.475
Matérn 3	-20.119	46.549
Matérn 5	-1.486	61.232

All covariance functions use automatic relevance determination. The Matérn covariance functions have regularity parameters 1, 3, and 5 respectively.

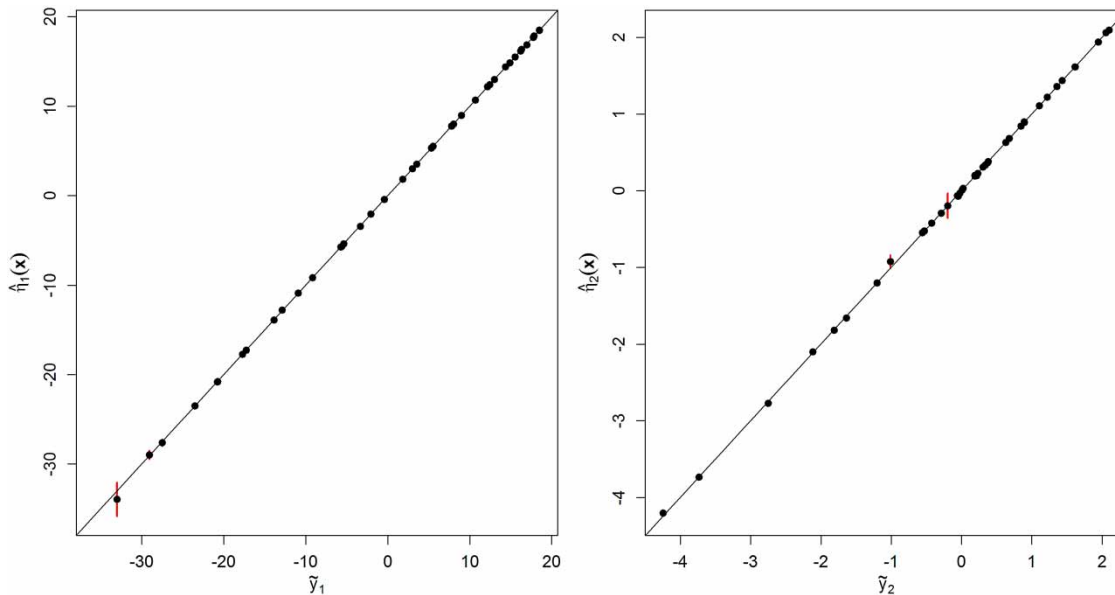


Figure 6 | Leave-one-out cross validation results for emulator $\hat{\eta}_k(\mathbf{x})$ of principal component \tilde{y}_k , $k = 1, 2$. Emulator mean predictions $\hat{\eta}_k(\mathbf{x})$ are plotted against principal component \tilde{y}_k (black dots). Emulator uncertainty is shown using 95% credible intervals (vertical red lines). Please refer to the online version of this paper to see this figure in colour: <http://dx.doi.org/10.2166/hydro.2019.067>.

marginal log-likelihood values for the Matérn covariance functions, which increase when the regularity parameter is larger.

Leave-one-out cross validation diagnostic plots for the two emulators $\hat{\eta}_k(\mathbf{x})$, $k = 1, 2$, are shown in Figure 6 with 95% credible intervals. It is evident that both emulators accurately predict the principal component output at the withheld design points. In particular, for both principal components the emulator's predictions lie on or very close to the $y = x$ line. In terms of coverage, for the first principal component two of the 40 withheld points lie outside of the emulator's credible intervals, giving a coverage of 95%. For the second principal component, four of the 40 withheld points lie outside of the emulator's credible intervals so the coverage is 90%. In summary, both emulators approximate the principal component output accurately and are well calibrated.

The emulators validate well on the space of the two principal component outputs. However, to be of any use, emulator predictions must be reconstructed and propagated back to the original space of the Rushton *et al.* (2006) model output: the daily river flow. A procedure for doing this was given above under 'Reconstruction and propagation of uncertainty'. It is useful to check whether the reconstructions accurately predict the daily river flow at unseen land

use inputs, and RMSE and MAE metrics are used to this end. Calculating the RMSE for each of the 40 leave-one-out cross validation points gives values ranging from 1.127×10^{-2} to 4.387×10^{-1} (with a mean of 8.604×10^{-2}). Similarly, the MAE for each of the 40 leave-one-out cross validation points gives values ranging from 4.570×10^{-2} to 1.322 (with a mean of 2.314×10^{-1}). The RMSE and MAE are measured in the same units as the model output (m^3/s). Given that the range of the Rushton *et al.* (2006) model output for the Frome basin is approximately between 0 and $25 \text{ m}^3/\text{s}$, these small values of the RMSE and MAE indicate a very good match between the simulated and emulated daily river flows. The difference in simulated and emulated daily river flow for the smallest and largest RMSE values is shown in Figure 7. The difference is plotted because it was hard to see a difference between the simulated and emulated time series on an absolute scale. For the case of the smallest RMSE value, the emulator predicts the model output almost exactly. For the largest RMSE value, the emulator mainly overestimates the daily river flow in a seasonal pattern, but the error is small and the uncertainty increases accordingly. It was found that larger values of the RMSE tended to correspond with flow output with larger magnitude, indicating that these

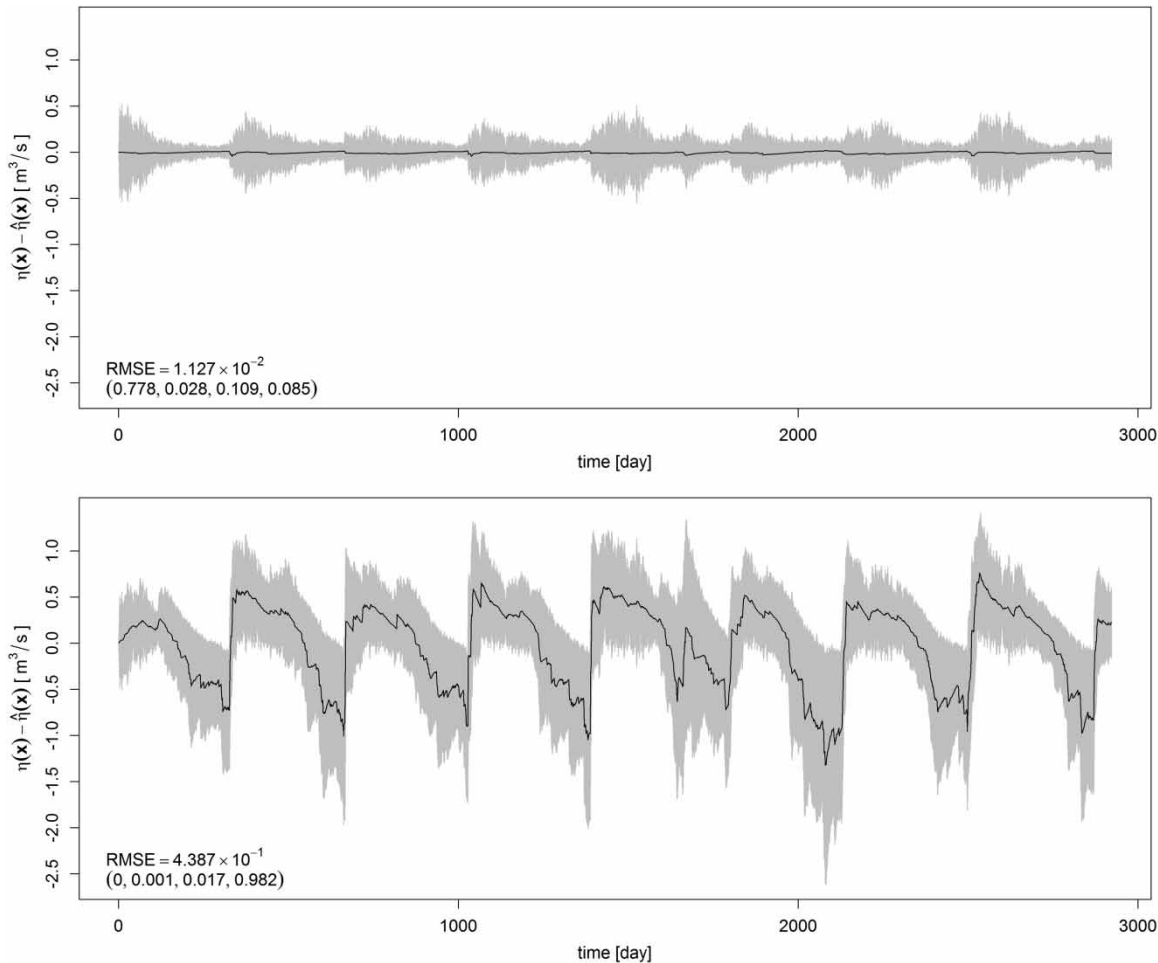


Figure 7 | Difference between simulated and emulated daily river flow, $\eta(\mathbf{x}) - \hat{f}(\mathbf{x})$, for the two leave-one-out cross validation land use inputs which gave the smallest (top panel) and largest (bottom panel) root square error (RMSE). For each panel, the RMSE and the land use input setting (x_1, x_2, x_3, x_4) is given.

outputs are harder to predict. However, this relationship was small and is not investigated further here.

Comparing the computational time of model and emulator

With the emulator validation complete, it is useful to investigate the reduction in computational burden it provides. The time it took for the model and emulator to run the 40 experiments in the CLHS design was recorded for 100 repetitions, with the average time for completion compared. In this work, experiments were run on a computer with an Intel Core i7 processor running at 3.4 GHz using 32 GB of RAM, on a Windows 7 operating system. The average completion time for the model and emulator was 8.768 and 2.182 seconds

respectively. This equates to approximately a four times speed up in computational time when using the emulator. This reduction is quite small but is not surprising considering that the Rushton *et al.* (2006) model is a fairly simple hydrological model, mainly used in this paper to demonstrate our emulation approach. Nevertheless, the difference in computational time would become substantial when performing tasks which require large numbers of simulations, or when using the emulator for more complex models.

CONCLUSIONS

Due to a rapid increase in the sophistication of environmental modelling, approaches to reduce the computational

time of the models have become popular. In this context, statistical emulators are used to provide fast approximations of computationally intensive models to facilitate subsequent analysis. With enough runs of the original model, emulators often offer predictions of comparable accuracy for a fraction of the computational time and, importantly, quantify the uncertainty associated with replacing the model with an emulator.

In this study, a novel emulator methodology has been proposed to provide a fast approximation to a hydrological model available in the literature. The model itself was reasonably simple and fast to run, but provided a good test example to demonstrate the emulator methodology. While emulators have been used for hydrological models before, the approach given here was novel in a number of ways, for example in its utilization and combination of a range of techniques from various fields. Firstly, principal component analysis was used to reduce the dimension of the time series output of the model. Secondly, due to the constrained nature of the land use inputs, the field of mixture experiments was drawn upon: classical mixture models were used alongside a novel mixture design strategy which used a CLHS algorithm to generate a space-filling design tailored for building statistical emulators. Finally, Gaussian process emulators were employed to describe the relationship between the land use inputs and the principal component outputs. The emulator's predictions were then reconstructed back to the space of the daily river flow output of the model. Furthermore, uncertainty due to the use of an emulator and dimension reduction techniques were efficiently propagated.

The statistical emulator presented here was applied to a version of the hydrological model calibrated for the Frome basin using measurements from the East Stoke Total station. Under 'Results and discussion' above it was shown first that the calibrated model provided a good description of the observed river flow at a daily and monthly scale, via the Nash and Sutcliffe coefficient. Second, the proposed statistical emulator gave very accurate predictions of the calibrated model output as a function of four land use inputs, as tested by a number of leave-one-out cross validation exercises and diagnostics. Possible future directions of this work would be to test the proposed methodology on hydrological models

calibrated to different river basins, or for more complex implementations of the model (for example with a larger number of land use inputs). It is anticipated that the approach outlined here would perform well in these scenarios.

DATA AVAILABILITY

Data and code for the hydrological model and emulator methodology are openly available at <https://github.com/neowen7/water-resource-emulation>.

REFERENCES

- Abrahart, R. J. & See, L. M. 2007 *Neural network modelling of non-linear hydrological relationships*. *Hydrol. Earth Syst. Sci. Discuss.* **11** (5), 1563–1579.
- Allen, R. G., Pereira, L. S., Raes, D. & Smith, M. 1998 *Crop Evapotranspiration – Guidelines for Computing Crop Water Requirements*. FAO Irrigation and Drainage Paper 56, 300(9), D05109. FAO, Rome.
- Brunner, G. W. 2001 *HEC-RAS, River Analysis System Hydraulic Reference Manual*. Technical report. U.S. Army Corps of Engineers, Hydrologic Engineering Center, Davis, CA.
- Carbajal, J. P., Leitão, J. P., Albert, C. & Rieckermann, J. 2017 *Appraisal of data-driven and mechanistic emulators of nonlinear simulators: the case of hydrodynamic urban drainage models*. *Environ. Model. Softw.* **92**, 17–27.
- Castelletti, A., De Zaia, M., Galelli, S., Restellia, B. M., Sanavia, P., Soncini-Sessa, R. & Antenucci, J. P. 2009 An emulation modelling approach to reduce the complexity of a 3D hydrodynamic-ecological model of a reservoir. *Environ. Softw. Syst.* **8**, 13–22.
- Castelletti, A., Pianosi, F., Soncini-Sessa, R. & Antenucci, J. P. 2010 *A multiobjective response surface approach for improved water quality planning in lakes and reservoirs*. *Water Resour. Res.* **46** (6), W06502.
- Castelletti, A., Galelli, S., Ratto, M., Soncini-Sessa, R. & Young, P. C. 2012 *A general framework for dynamic emulation modelling in environmental problems*. *Environ. Model. Softw.* **34**, 5–18.
- Christelis, V., Regis, R. G. & Mantoglou, A. 2017 *Surrogate-based pumping optimization of coastal aquifers under limited computational budgets*. *J. Hydroinform.* **20** (1), 164–176.
- Constantine, P. G. 2015 *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*. SIAM, Philadelphia, PA.
- Cornell, J. A. 2011 *Experiments with Mixtures: Designs, Models, and the Analysis of Mixture Data*. John Wiley & Sons, New York.

- Craig, P., Goldstein, M., Seheult, A. & Smith, J. 1996 Bayes linear strategies for matching hydrocarbon reservoir history. In: *Bayesian Statistics*, Vol. 5 (J. Bernardo, J. Berger, A. Dawid & A. Smith eds). Oxford University Press, Oxford, UK, pp. 69–95.
- DHI 1997 *MIKE11 GIS Reference and User Manual*. Danish Hydraulic Institute, Horsholm, Denmark.
- Feranec, J., Soukup, T., Hazeu, G. & Jaffrain, G. 2016 *European Landscape Dynamics: CORINE Land Cover Data*. CRC Press Taylor & Francis Group, Boca Raton, FL, USA; London, UK; New York, NY, USA.
- Fraser, C. E., McIntyre, N., Jackson, B. M. & Wheeler, H. S. 2013 Upscaling hydrological processes and land management change impacts using a metamodeling procedure. *Water Resour. Res.* **49** (9), 5817–5833.
- Fu, G., Makropoulos, C. & Butler, D. 2010 Simulation of urban wastewater systems using artificial neural networks: embedding urban areas in integrated catchment modelling. *J. Hydroinform.* **12** (2), 140–149.
- Haylock, R. G. & O'Hagan, A. 1996 On inference for outputs of computationally expensive algorithms with uncertainty on the inputs. *Bayesian Statist.* **5**, 629–637.
- Ireson, A. M. & Butler, A. P. 2013 A critical assessment of simple recharge models: application to the UK chalk. *Hydrol. Earth Syst. Sci.* **17** (6), 2083–2096.
- Jolliffe, I. T. 2002 *Principal Component Analysis*. Springer, New York, NY.
- Katurji, M., Nikolic, J., Zhong, S., Pratt, S., Yu, L. & Heilman, W. E. 2015 Application of a statistical emulator to fire emission modeling. *Environ. Model. Softw.* **73**, 254–259.
- Kennedy, M. C. & O'Hagan, A. 2001 Bayesian calibration of computer models. *J. R. Stat. Soc. B Stat. Methodol.* **63**, 425–464.
- Leeds, W. B., Wikle, C. K., Fiechter, J., Brown, J. & Milliff, R. F. 2013 Modeling 3-D spatio-temporal biogeochemical processes with a forest of 1-D statistical emulators. *Environmetrics* **24** (1), 1–12.
- Lima, G. M., Brentan, B. M., Manzi, D. & Luvizotto, E. 2018 Metamodel for nodal pressure estimation at near real-time in water distribution systems using artificial neural networks. *J. Hydroinform.* **20** (2), 486–496.
- Loeppky, J. L., Sacks, J. & Welch, W. J. 2009 Choosing the sample size of a computer experiment: a practical guide. *Technometrics* **51**, 366–376.
- Machac, D., Reichert, P., Rieckermann, J. & Albert, C. 2016 Fast mechanism-based emulator of a slow urban hydrodynamic drainage simulator. *Environ. Model. Softw.* **78**, 54–67.
- McKay, M. D., Beckman, R. J. & Conover, W. J. 1979 Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245.
- Minasny, B. & McBratney, A. B. 2006 A conditioned Latin hypercube method for sampling in the presence of ancillary data. *Comput. Geosci.* **32**, 1378–1388.
- Mirfenderesgi, G. & Mousavi, S. J. 2015 Adaptive meta-modeling-based simulation optimization in basin-scale optimum water allocation: a comparative analysis of meta-models. *J. Hydroinform.* **18** (3), 446–465.
- Moonen, P. & Allegrini, J. 2015 Employing statistical model emulation as a surrogate for CFD. *Environ. Model. Softw.* **72**, 77–91.
- Nash, J. E. & Sutcliffe, J. V. 1970 River flow forecasting through conceptual models part I – a discussion of principles. *J. Hydrol.* **10** (3), 282–290.
- O'Hagan, A. 2006 Bayesian analysis of computer code outputs: a tutorial. *Reliability Eng. Syst. Safety* **91**, 1290–1300.
- Rasmussen, C. E. & Williams, C. K. I. 2006 *Gaussian Process for Machine Learning*. The MIT Press, Cambridge, MA.
- Razavi, S., Tolson, B. A. & Burn, D. H. 2012 Review of surrogate modeling in water resources. *Water Resour. Res.* **48** (7), W07401.
- R Core Team 2017 *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. Available from: www.R-project.org/.
- Ritter, A. & Muñoz-Carpena, R. 2013 Performance evaluation of hydrological models: statistical significance for reducing subjectivity in goodness-of-fit assessments. *J. Hydrol.* **480**, 33–45.
- Romanowicz, R. J. & Osuch, M. 2015 Stochastic Transfer Function based emulator for the on-line flood forecasting. In: *Stochastic Flood Forecasting System* (R. J. Romanowicz & M. Osuch, eds). Springer, Cham, pp. 159–170.
- Romanowicz, R. J., Beven, K. J. & Tawn, J. 1996 Bayesian calibration of flood inundation models. In: *Floodplain Processes* (M. G. Anderson & D. E. Walling, eds). Wiley, Chichester, pp. 333–360.
- Roudier, P. 2011 *clhs: A R Package for Conditioned Latin Hypercube Sampling*. R Foundation for Statistical Computing. Available from: <https://cran.r-project.org/web/packages/clhs/clhs.pdf> (accessed June 2018).
- Roustant, O., Ginsbourger, D. & Deville, Y. 2012 Dicekriging, DiceOptim: two R packages for the analysis of computer experiments by Kriging-based metamodeling and optimization. *J. Stat. Softw.* **51**, 1–55.
- Roy, T., Schütze, N., Grundmann, J., Brettschneider, M. & Jain, A. 2016 Optimal groundwater management using state-space surrogate models: a case study for an arid coastal region. *J. Hydroinform.* **18** (4), 666–686.
- Rushton, K. R., Eilers, V. H. M. & Carter, R. C. 2006 Improved soil moisture balance methodology for recharge estimation. *J. Hydrol.* **318**, 379–399.
- Sammut, C. & Webb, G. I. 2017 *Encyclopedia of Machine Learning and Data Mining*. Springer Science & Business Media, New York.
- Scheffé, H. 1958 Experiments with mixtures. *J. R. Stat. Soc. B Method.* **20**, 344–360.
- Solomatine, D. P. & Ostfeld, A. 2008 Data-driven modelling: some past experiences and new approaches. *J. Hydroinform.* **10** (1), 3–22.
- Stanfill, B., Mielenz, H., Clifford, D. & Thorburn, P. 2015 Simple approach to emulating complex computer models for global sensitivity analysis. *Environ. Model. Softw.* **74**, 140–155.

- Urban, N. M. & Fricker, T. E. 2010 [A comparison of Latin hypercube and grid ensemble designs for the multivariate emulation of an Earth system model](#). *Comput. Geosci.* **36**, 746–755.
- Wilkinson, R. D. 2011 Bayesian calibration of expensive multivariate computer experiments. In: *Large-scale Inverse Problems and Quantification of Uncertainty, Chapter 10* (L. Beigler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, Y. Marzouk, L. Tenorio, B. van Bloemen Waanders & K. Willcox, eds). John Wiley & Sons, New Jersey, USA, pp. 195–215.
- Williamson, D., Blaker, A. T., Hampton, C. & Salter, J. 2015 [Identifying and removing structural biases in climate models with history matching](#). *Clim. Dynam.* **45**, 1299–1324.
- Young, P. C., Leedal, D. & Beven, K. J. 2009 Reduced order emulation of distributed hydraulic models. In: *Proceedings 15th IFAC Symposium on System Identification SYSID09*, St. Malo, France.

First received 6 July 2018; accepted in revised form 13 February 2019. Available online 19 March 2019