Environmental Modelling & Software 76 (2016) 1–12

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

A GUI platform for uncertainty quantification of complex dynamical models

Chen Wang [a], Qingyun Duan [a, *], Charles H. Tong [b], Zhenhua Di [a], Wei Gong [a]

[a] College of Global Change and Earth System Science and Joint Center for Global Change Research, Beijing Normal University, Beijing, 100875, China
[b] Lawrence Livermore National Laboratory, 7000 East Ave, Livermore, CA, 94550, USA

ARTICLE INFO

Article history:
Received 25 March 2015
Received in revised form
11 November 2015
Accepted 13 November 2015
Available online xxx

Keywords:
Uncertainty Quantification
Design of experiments
Sensitivity analysis
Surrogate modeling
Parameter optimization
UQ-PyL

ABSTRACT

Uncertainty quantification (UQ) refers to quantitative characterization and reduction of uncertainties present in computer model simulations. It is widely used in engineering and geophysics fields to assess and predict the likelihood of various outcomes. This paper describes a UQ platform called UQ-PyL (Uncertainty Quantification Python Laboratory), a flexible software platform designed to quantify uncertainty of complex dynamical models. UQ-PyL integrates different kinds of UQ methods, including experimental design, statistical analysis, sensitivity analysis, surrogate modeling and parameter optimization. It is written in Python language and runs on all common operating systems. UQ-PyL has a graphical user interface that allows users to enter commands via pull-down menus. It is equipped with a model driver generator that allows any computer model to be linked with the software. We illustrate the different functions of UQ-PyL by applying it to the uncertainty analysis of the Sacramento Soil Moisture Accounting Model. We will also demonstrate that UQ-PyL can be applied to a wide range of applications.

Software availability

Name of software: Uncertainty Quantification Python Laboratory (UQ-PyL)
Programming language: Python
Operating system: Windows, Linux and MacOS
Availability: http://www.uq-pyl.com
Documentation: http://www.uq-pyl.com
User interface: Graphical user interface or command-line
License: Free under a GNU General Public License (www.gnu.org) agreement.

1. Introduction

Computer models have become an indispensable tool in engineering and geophysics for simulating the behaviors of complex dynamical systems. People can do things with computer models which may otherwise be impossible in real life, e.g., the simulation of nuclear explosions and the prediction of future climatic and hydrologic events (Ji et al., 2014). However, computer simulations inherently involve uncertainties (Box and Draper, 1987). Quantifying those uncertainties is critical to assess the ability of computer models to emulate actual system behavior. Uncertainty quantification (UQ) refers to quantitative characterization and reduction of uncertainties present in computer model simulations. It tries to determine how likely certain outcomes are given that some aspects of the system are not exactly known. Tong (2005) has defined the objectives of UQ as: 1) to characterize the output uncertainties of a simulation model, 2) to identify the major sources of uncertainties of a model, and to provide information on which additional experiments are needed to improve the understanding of a model, and 3) to tune a simulation model to match model outputs to experiments.

The errors leading to discrepancies between model simulations and observed experimental data can be grouped into three categories (Duan et al., 2006): 1) model error, 2) data error, and 3) parameter error. All computer models are only an abstraction of the real world. They are usually designed to express only essential principles of the systems (e.g. conservation laws, thermodynamics laws, etc.) (Maitre and Knio, 2010). Often, simplifications are made to allow for mathematical tractability and/or to compensate for our lack of knowledge of certain processes. Therefore, all models contain model errors. A multi-model ensemble strategy is sometimes used as a practical way to account for uncertainties due to

* Corresponding author.
E-mail address: qyduan@bnu.edu.cn (Q. Duan).

http://dx.doi.org/10.1016/j.envsoft.2015.11.004

model errors (Du, 2007). All computer models need input data to conduct a simulation. The data may concern the system's geometry, boundary and initial conditions and external forcing. Using data which is only an imperfect approximation of an environmental variable introduces additional errors known as data errors. In practice, different data assimilation algorithms have been advanced to address data errors (Houtekamer and Mitchell, 1998). All computer models come with parameters that reflect physical characteristics of the simulated system. These parameters may be physical constants or coefficients and exponents in equations prescribing the constitutive laws of the system. Incorrect specification of the parameter values (i.e., parameter errors) can have great influence on the performance of the simulation model (Duan et al., 2006). In this study, we focus on quantifying and reducing uncertainties due to parameter errors.

In traditional hydrological modeling, model parameters are often estimated through model calibration, a process of matching model simulation to experimental data by tuning model parameters with some optimization algorithms (Duan et al., 1992). However, an optimization search may need up to tens of thousands of model runs to find the global optimal solution (Wang et al., 2014). This can be problematic if a computer model is sufficiently complex with a high dimensionality and a high computational demand. For this kind of models, a UQ framework can be used. UQ framework includes some key steps to simplify the problem and reduce computational demand. If a model has a large number of parameters, but only a relatively small number of them are important, a screening procedure can be used to identify the most important ones. Sensitivity analysis (SA) is a tool designed to accomplish this task (Shin et al., 2013; Li et al., 2013; Gan et al., 2014; Shin et al., 2015). If the simulation model requires a high computational resource to run, surrogate modeling should be used by constructing a simple statistical model of the response surface of the simulation model (Wang et al., 2014). Once the surrogate model is constructed, a global optimization algorithm can be used to identify the optimal parameter set.

A UQ software platform is designed to accomplish the tasks required for a UQ framework. A good UQ software platform should have the following features: 1) it should have many different UQ algorithms for users to choose from and should have a good architecture for extending new algorithms; 2) it should have a feature that allows easy coupling with any user-defined external models; 3) the platform can run across different platforms, including Windows, Linux and MacOS and 4) it should have a Graphical User Interface (GUI) which is convenient to use for the users.

There exist a plethora of different UQ software platforms, each of them having unique strengths and weaknesses and being suitable for different kind of problems. PEST (Parameter ESTimation), which runs under the Windows platform, is a model-independent parameter estimation software for complex environmental and other computer models (Doherty, 2004). PSUADE (Problem Solving environment for Uncertainty Analysis and Design Exploration) is a C++ based open-source software package, which provides an integrated design and analysis environment for performing UQ analyses of large complex system models (Tong, 2005). It runs on Linux based systems and only through command line languages. DAKOTA (Design Analysis Kit for Optimization and Terascale Application) toolkit, developed by researchers from Sandia National Laboratories, is written in C++ and runs on supercomputer platforms (Adams et al., 2009). It provides a flexible and extensible interface between simulation codes and the iterative analysis methods. SIMLAB[1] (Saltelli et al., 2004) is a Windows based software platform which mainly focuses on sensitivity analysis. It provides a reference implementation of many global sensitivity analysis techniques. GUI-HDMR (Ziehn and Tomlin, 2009) is also a sensitivity analysis software. It has a GUI and can only run with MATLAB platform. UCODE performs inverse modeling, posed as a parameter estimation problem, using nonlinear regression approach (Poeter et al., 2005). It is written in FORTRAN language and can only run on a Windows platform. Open TURNS (Open source initiative to Treat Uncertainties, Risks'N Statistics) which is a powerful platform to perform uncertainty and sensitivity analysis (Andrianov et al., 2007). It can run on Windows and Linux based operating systems and need Python language to call its functions. UQLab (Uncertainty Quantification in MATLAB) is a MATLAB based software framework for uncertainty quantification which is designed to be extended to the engineering research community (Marelli and Sudret, 2014). It has incorporated mainly the non-intrusive stochastic methods, such as Polynomial Chaos Expansion (PCE). Of those UQ platforms, some focus on specific aspects of the UQ process, e.g., PEST designed mainly to solve optimization problems and SIMLAB to perform only uncertainty and sensitivity analyses. Some of the software run only on a particular operating system, e.g., PEST, UCODE and SIMLAB running only on Windows platform, PSUADE only on Linux platform. Some need third party commercial software, e.g., UQLab needing to run under MATLAB software. Few of them contain comprehensive toolsets for carrying out the entire UQ process, run on cross-platforms, and have a GUI to use.

In this study, we developed a new, general-purpose, cross-platform UQ framework called Uncertainty Quantification Python Laboratory (UQ-PyL), which aims to combine the strengths of several existing UQ platforms, while also serving as a tutorial toolkit for UQ users. It is a Python based software platform designed to help computer modelers to quantify and reduce model uncertainties associated with model parameters. It is made of several components that perform various functions, including design of experiments, statistical analysis, sensitivity analysis, surrogate modeling and parameter optimization. It is suitable for parametric uncertainty analysis of any computer simulation models, as long as three sets of model related information are available: 1) the model executable code with all required input forcing data files, 2) the control file which contains all information needed to execute the model (e.g., all adjustable parameters and file access information for all inputs and outputs), and 3) model simulation outputs of interest, the corresponding observations and the error function. UQ-PyL is intended as a didactic as well as a practical toolbox, and therefore, it contains many different methods under each function. It runs on common computer system platforms including Windows, Linux and MacOS and has a GUI that helps execute each of the functions.

The purpose of this paper is to introduce UQ-PyL to users, and provide useful guidance on using appropriate UQ functions for their applications. The paper is organized as follows: Section 2 offers a brief description of UQ framework. Section 3 describes the UQ-PyL platform. Section 4 illustrates the different functions of UQ-PyL through the calibration of a simple conceptual hydrologic model – Sacramento Soil Moisture Accounting (SAC-SMA) model (Burnash, 1995). Finally, concluding remarks and future work are presented in Section 5.

2. Framework of Uncertainty Quantification

To rigorously perform UQ, a comprehensive methodology is needed. Fig 1 shows the flowchart of the UQ procedure. The main steps involved in UQ include defining the problem, characterizing the uncertainties, screening the most sensitive parameters, constructing a suitable surrogate model and performing analyses such

[1] http://ipsc.jrc.ec.europa.eu/index.php?id=756#c2907.

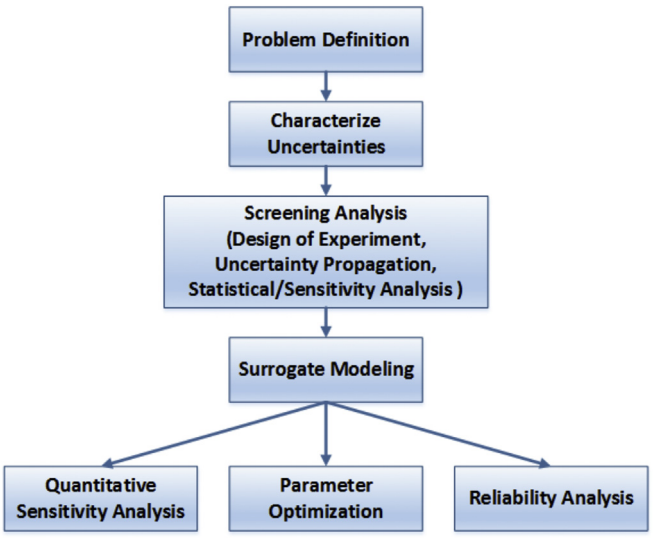

Fig. 1. Flowchart of Uncertainty Quantification.

as quantitative global sensitivity analysis, global optimization or reliability analysis. Below, we elaborate on each of those steps.

First, the user should define the problem that needs to be solved. In this step, the suitable computer model of a physical system should be selected, along with the input and output variables clearly specified. The types of input data generally depend on the physical problem being simulated, and on the model used to represent it.

The initial characterization of parameter uncertainties should be based on prior knowledge of the model. By characterizing a source of uncertainty we mean that a) assigning a mathematical structure to describe the uncertainty and b) determining the numerical values of all of the relevant parameters of the structure. In this study, the input variables are specified including parameter name, lower and upper bound and functional form of the probability distribution.

If the model has a high parameter dimensionality, a screening step should be performed to screen out the less sensitive parameters from the more sensitive ones in order to reduce the dimensionality. Screening analysis step includes several components: design of experiment, uncertainty propagation and statistical/sensitivity analysis. Design of experiment (DoE) is a body of techniques that enable an investigator to conduct better experiments, analyze data efficiently and make the connections between the conclusions from the analysis and the original objectives of the investigation (Wu and Hamada, 2009). DoE can be divided into deterministic design, random design and quasi-random design. Deterministic design means that if the number of factors and the levels in each factor are known, the number of samples and the position of each sample are determined, e.g., Full-Factorial design (Box et al., 2005). Random design refers to the design which generates a specified number of samples randomly according to a predetermined probability distribution, e.g., Monte Carlo design. Quasi-random design, also referred as "low discrepancy design", denotes the notion that the parameter samples are not really random, but are generated according to some schemes that place them in sample space that appear more uniformly distributed than a random design. In other words, the sample points in quasi-random design appear random, but the exact location of every sample point is set deterministically. A typical example is Sobol' sequence which is generated using a base of two to form successively finer uniform partitions of the unit interval, and then reorder the coordinates in each dimension (Sobol', 1967). After performing DoE, uncertainty propagation should be followed by running the simulation model using the parameter sample sets that were generated.

A statistical analysis of the model outputs can be performed after model simulations are completed for all of the DoE generated parameter sample sets. Statistical analysis include calculation of various statistical moments (e.g., means, standard deviations, skewness and kurtosis), confidence intervals of the estimated means, correlational analysis between model parameters and model performance measures, and hypotheses tests on output statistics and distributions.

Sensitivity analysis (SA) refers to the process of how uncertainty in the output of a model can be apportioned to different input factors, i.e., model parameters (Saltelli et al., 2008). SA can be divided into two groups: local SA and global SA. The local SA measures the changes of model response by varying only one parameter at a time while keeping other parameters constant, while global SA examines the changes of model response by varying all parameters sequentially or simultaneously (Gan et al., 2014). For global SA, it can be categorized into derivative-based approach, variance-based approach and regression-based approach. The principle of derivative-based approach is that the derivative $\partial Y/\partial X_i$ of an output Y versus an input parameter X_i can be thought of as a mathematical definition of the sensitivity of Y versus X_i (Saltelli et al., 2008). The larger the value of $\partial Y/\partial X_i$, the more sensitive is the input parameter X_i is. For variance-based method, the sensitivity of an input parameter is determined based on its contribution to the model output total variance. The method is founded on the idea that if D is the total variance of the model output, this can be decomposed in terms of increasing dimensionality as $D = \sum_{i=1}^{n} D_i + \sum \sum_{1 \le i \le j \le n} D_{ij} + \cdots + D_{12\cdots n}$, where $D_{(\ldots)}$ denotes the variance contributed by the terms in (…), with a bigger value implying higher sensitivity. Note that in theory variance decomposition method can not only offer the first order sensitivity information, but also higher order information and the total variance information. Regression-based SA methods may also be called response surface SA methods, a class of methods which determine the sensitivity of an input factor by its ability to influence the quality of regression or response surface. Multivariate adaptive regression splines (MARS) method is a typical example of this type of methods, which determines the sensitivity based on difference between the response surface established using all input factors and the one which is built without a specified input factor. The larger the difference between the response surface, the more sensitive is the input factor (Li et al., 2013; Gan et al., 2014).

If a simulation model requires a significant computational resource to run, surrogate modeling should be used to construct a simple statistical emulator of the response surface of the dynamical model. Surrogate modeling is a collection of mathematical and statistical techniques for empirical model building. Surrogate modeling methods describe the relationship between inputs (i.e., model's adjustable parameters) and outputs (i.e., the performance measure of the dynamical simulation model) (Wang et al., 2014). Once a surrogate model has been constructed, subsequent analysis can rely on this surrogate model, which is inexpensive to run compared to the dynamical simulation model. This should facilitate quantitative analysis that requires a large number of model runs. Some fields also refer surrogate modeling methods as function approximation, meta-modeling, response surface method or statistical emulation.

Once the surrogate model is established, comprehensive analyses requiring up to tens of thousands model runs, such as quantitative sensitivity analysis, global optimization, and reliability

analysis, can be performed on the surrogate model. In this study, we focus on parameter optimization, also called model calibration which is essentially a global optimization problem (i.e., given a set of tunable parameters and their ranges, it searches for optimal parameter set that yields model simulation that best matches with the experimental data). Two kinds of optimization approaches can be employed for model calibration, namely, deterministic optimization which tries to identify a unique set of optimal parameters and Bayesian optimization which aims to find the posterior distribution of the parameter sets that maximize of the likelihood function given the observations.

3. An overview of the UQ Python Laboratory (UQ-PyL) platform

3.1. A description of the UQ-PyL

The UQ-PyL software platform is developed to provide an integrated design and analysis environment for performing various UQ tasks for large complex simulation models. It has assembled many of the most commonly used UQ algorithms and makes them easily available in a single platform. The software is written in Python language, one of the most popular languages for academic research and industrial applications. UQ-PyL makes use of numerous external Python packages to perform various functions. For example, Numpy[2] package is used to define the basic data structure. Scipy[3] package is incorporated for its efficient algorithms for linear algebra, sparse matrix representation and basic statistical functions. Matplotlib[4] package is utilized for its various drawing functions, and Scikit-learn (Pedregosa et al., 2011) package is integrated into UQ-PyL for the different surrogate modeling algorithms. UQ-PyL also integrates some existing packages to perform certain uncertainty quantification functions, e.g., pyDoE[5] package for carrying out DoE functions, SALib[6] package for implementing various sensitivity analysis algorithms.

Even though UQ-PyL is equipped with a GUI to facilitate execution of various functions, it can also run as a script program in a batch mode. Designed to run over different platforms, UQ-PyL can generate both Python script and shell script which can run on a Windows (with *.bat format), a Linux, or a MacOS platform (with *.sh format). One may create a script file on a Windows platform and then port it into other platforms.

Fig. 2 shows the front page of UQ-PyL. Different tab widgets allow the user to execute different steps of the UQ process, including problem definition, DoE, statistical analysis, SA, surrogate modeling and parameter optimization. One may click on the desired tab by using a mouse and/or by entering the required information via keyboard to perform various tasks. After a task is completed, the software generates tabular and graphical outputs. The graphical outputs can be saved in a variety of formats, including.png,.bmp,.tiff or.pdf among others. UQ-PyL also has an interactive interface realized through Spyder package.[7] This feature allows users to perform various functions in UQ-PyL through interactive execution of Python scripts (see Fig. 3). The left of the interactive window is a Python script editor, in which one may edit the Python codes that contain various UQ-PyL intrinsic functions. In this window, one may click the "run" button to execute the script. The values of the internal variables are shown in the upper right and the UQ output results are shown in the lower right of the window.

[2] http://www.numpy.org/.
[3] http://www.scipy.org/.
[4] http://matplotlib.org/index.html.
[5] http://pythonhosted.org/pyDOE/.
[6] http://jdherman.github.io/SALib/.
[7] http://pythonhosted.org/spyder/.

3.2. The UQ-PyL flowchart

Fig. 4 is the flowchart illustrating how UQ-PyL executes an UQ task. A task can be carried out in three major steps: (1) model configuration preparation; (2) uncertainty propagation; and (3) UQ analysis. In the first step, the GUI is used to specify the model configuration information (i.e., parameter names, ranges and distributions), and the DoE information (i.e., the sampling techniques and sample sizes) to prepare for UQ exercise for a given problem. At the end of this step, a file called "sample set" is generated which contains all information on model configuration and the DoE sampling results. In the second step, the different sample parameter sets generated in the last step are fed into the simulation model to enable the execution of simulation model (i.e., uncertainty propagation). To do this, UQ-PyL generates a "model driver" (i.e., a Python script) which links model executable code with the "sample set" file. The "model driver" is then executed by running the simulation model according to the specified model configuration using different parameter sets as in the "sample set" file. At the end of this step, the "model response outputs" file is obtained, which contains all simulation response outputs to different parameter sets. In the third step, a variety of UQ exercises are carried out, including statistical analysis, SA, surrogate modelling and parameter optimization. UQ-PyL contains a rich set of tools for carrying out those exercises (see Tables 1–5). At the end of this step, tabular results and graphical outputs are obtained. For more detailed instructions on how to use UQ-PyL, readers are referred to the UQ-PyL online user manual (http://uq-pyl.com/file/UQ-PyL User Manual v1.1.pdf).

4. An illustration of UQ-PyL with a hydrological modeling example

4.1. The test model and data

This section intends to illustrate different UQ functions available in UQ-PyL. The SAC-SMA rainfall-runoff model is used as a test problem which has a highly non-linear, non-monotonic input parameter-model output relationship. This model is the most widely used hydrological model by the River Forecast Centers (RFCs) of the U.S. National Weather Service for catchment modeling and flood forecasting (Burnash et al., 1973). There are sixteen parameters in the SAC-SMA model. Thirteen of them are considered tunable, and the other three parameters are fixed at pre-specified values according to Brazil (1988). Table 6 describes those parameters and their ranges.

The South Branch Potomac River basin near Springfield, West Virginia in the U.S. was chosen as the study area. The total drainage area upstream of the gauging station (U.S. Geological Survey Station No. 01608500) is about 3800 km^2. Historical precipitation, potential evapotranspiration and streamflow observations from January 1, 1960 to December 31, 1979 were obtained from the Model Parameter Estimation Experiment (MOPEX) database for this study (Duan et al., 2006). The average annual precipitation over this period is 1021 mm, average annual potential evapotranspiration is 762 mm, and average annual runoff is 39.5 m^3/s. The hydrological simulations were run at a 6-h time step over the entire data period. To evaluate model response to different parameters, we use root mean square error (RMSE) between the simulated and observed daily streamflow discharge (m^3/s) as the objective function:

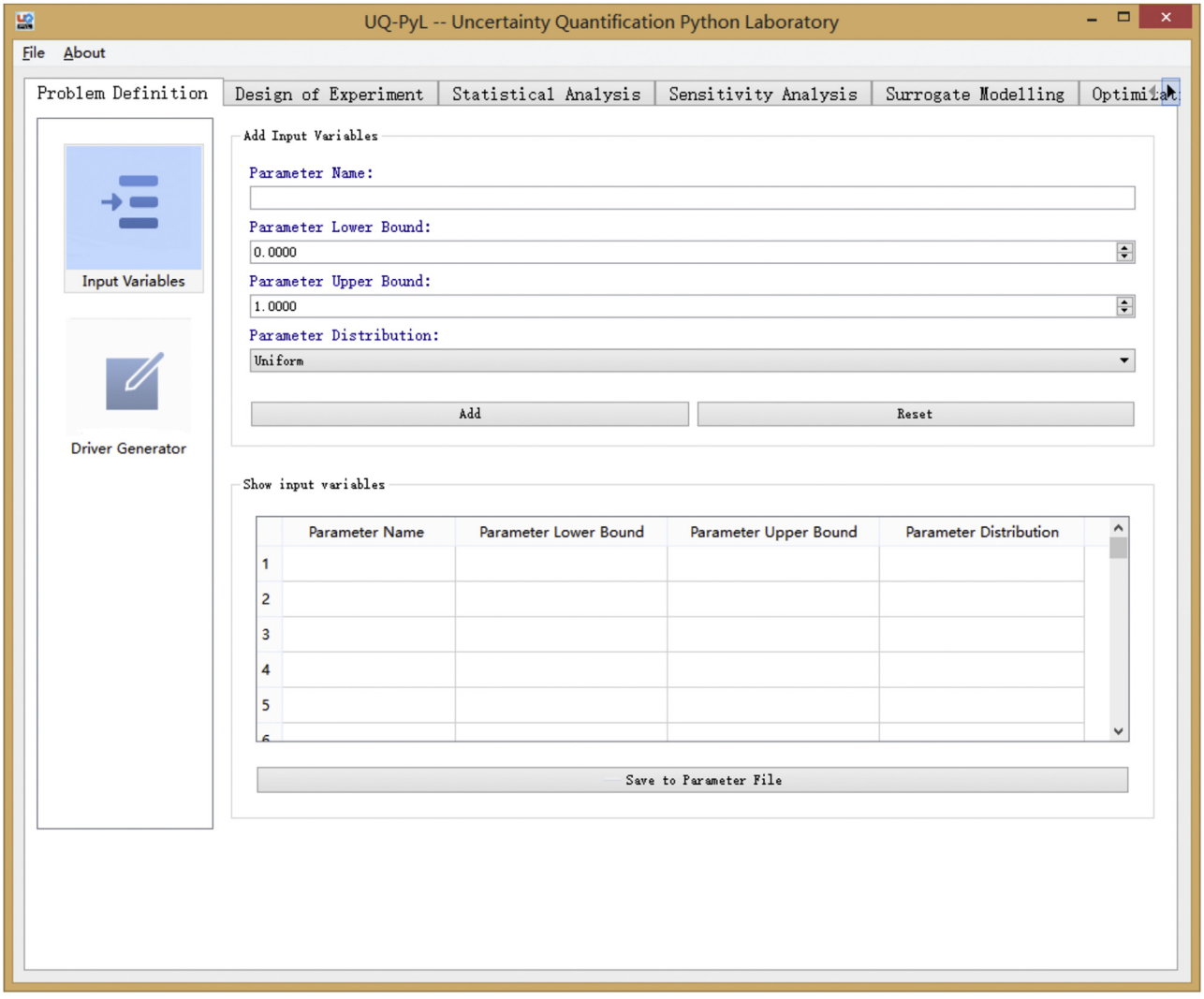

Fig. 2. Graphical user interface of UQ-PyL.

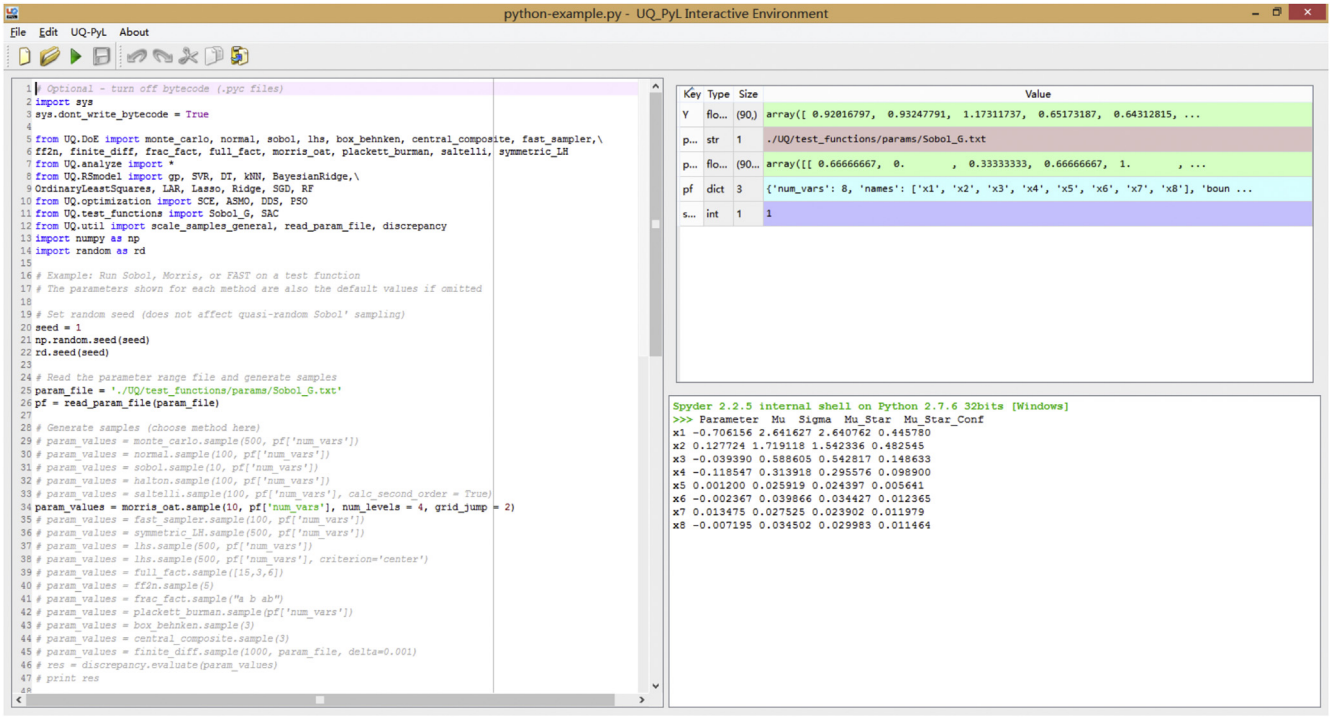

Fig. 3. Interactive page of UQ-PyL.

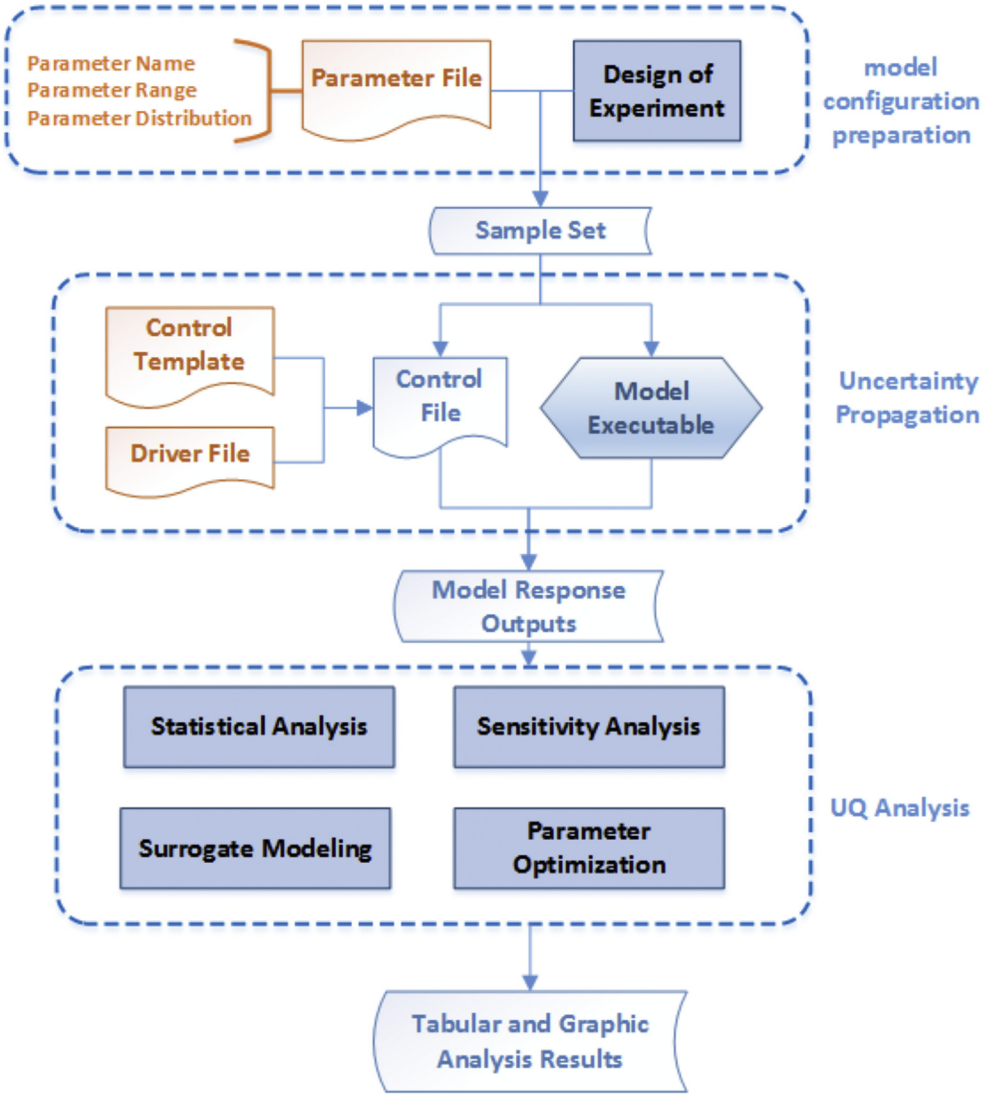

Fig. 4. UQ-PyL flowchart.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(Q_t^{fcs} - Q_t^{obs}\right)^2} \quad (1)$$

where Q_t^{fcs} and Q_t^{obs} are simulated and observed streamflow discharge values at time t, N is the total number of observations. To reduce the influence of incorrect specification of initial conditions, the simulations from the first three months are excluded in the calculation. The UQ-PyL GUI was used to generate a Python driver file to couple the SAC-SMA model with the software platform.

4.2. A demonstration of the DoE methods

DoE is the first step of UQ analysis. There are many DoE algorithms available in UQ-PyL, three of them are demonstrated here:

Table 1
DoE methods available in UQ-PyL.

Design type	Design method	Source
Deterministic	Full-Factorial design	Box et al., 2005
	Fractional-Factorial design	Box et al., 2005
	Plackett-Burman design	Plackett and Burman, 1946
	Box-Behnken design	Box and Behnken, 1960
	Central-Composite design	Myers, 1971
Random	Monte Carlo design	Meteopolis and Ulam, 1949
	Latin Hypercube design	Mckay et al., 1979, Iman et al., 1981
	Symmetric Latin Hypercube design	Ye et al., 2000
Quasi-random	QMC Sobol' sequence	Sobol', 1967
	QMC Halton sequence	Krykova, 2003

Table 2
Statistical Analysis methods available in UQ-PyL.

UA method	Source
Moment method	Casella and Berger, 2002
Confidence interval methods	Cox and Hinkley, 1974
Hypothesis test	Schervish, 1996
Analysis of variance (ANOVA)	Schervish, 1996

Latin Hypercube (LH), Monte Carlo (MC) and Quasi Monte Carlo (QMC). LH is a statistical method for generating a distribution of plausible collections of parameter values from a multi-dimensional distribution. It is a sampling design suited for an arbitrary number of dimensions, where each sample is the only one in each axis-aligned hyper-plane containing it. MC is the most commonly used algorithm that relies on repeated random sampling to obtain numerical approximations of a specified distribution function of an unknown probabilistic entity. A QMC can be viewed as a deterministic version of MC (Niederreiter, 1992). It also called a low-discrepancy sequence which is a set of points filling the sample area "efficiently" and has a lower discrepancy than straight pseudo-random number set (Krykova, 2003).

For the 13-dimensional SAC-SMA problem, we use 500 sample points to evaluate the four methods. We repeat LH and MC ten times. Since QMC is a deterministic sequence, it is run just one time. Four metrics MD2, CD2, SD2 and WD2 (Gong et al., 2015) were computed to evaluate the space-filling properties of these methods. For those metrics, the lower the numerical value, the better space-filling property the DoE algorithm has. Fig. 5 shows the space-filling comparison results of the four DoE algorithms. From the results we can see that QMC has the best score among four DoEs. For the remainder of this study, we choose QMC method for subsequent analyses.

4.3. The demonstration of statistical analysis and SA methods

In this section, we perform statistical analyses of the model outputs using the parameter samples generated by the QMC method from the previous section. The uncertainty of the model outputs can be characterized by using a few basic statistical quantities, including moments and confidence interval. Moments refer to statistical measures of a random variable Y, in this case the model output. The collection of all the moments uniquely determines the variable's probability distribution. In UQ-PyL, we automatically compute the first moment (mean), the second moment (variance), the third moment (skewness) and the fourth moment (kurtosis). The results are presented in Table 7. The confidence interval of the estimated mean is shown in Table 8, which measures uncertainty of the mean estimate due to sampling errors (Easton and McColl, 1997).

We also experimented with two of the SA methods from the UQ-PyL: the Morris One-at-A-Time (MOAT) and the Metamodel-based Sobol' variance decomposition method. The MOAT method is one of the most popular methods for parameter screening for its simplicity and efficiency (Morris, 1991). The theoretical basis of this method is that the overall effect of each parameter can be approximated by the mean μ and standard deviation δ of the gradients of each parameter sampled from the Morris sample paths. The gradients are the difference of objective function values divided by the difference of sampled parameter values. The MOAT method is a qualitative sensitive analysis method which gives only relative sensitivity results of different parameters. Sobol' variance decomposition method is a quantitative SA algorithm as it computes the precise contribution ratio of each parameter to the total variance of model output. However, the Sobol' method needs tens of thousands of model runs to obtain accurate approximation of the variances. This can be very difficult if the simulation model is expensive to run. UQ-PyL implemented a method called Metamodel-based Sobol' method, which calculates Sobol' variance decomposition efficiently based on the response surface of a metamodel. This method computes several SA indices including the first order sensitivity, total order sensitivity and second order interaction sensitivity.

Figs. 6 and 7 show the SA results using the MOAT and Metamodel-based Sobol' methods. For the MOAT method, the larger the modified means, the more sensitive the parameters are. The rule of thumb for delineating sensitive parameters from insensitive ones can be the sensitivity ratio of the most sensitive parameter vs the least sensitive one being less than a certain threshold (say 15–20). Based on the MOAT results, we observe that parameters LZPK, PCTIM, ADIMP, LZSK, PFREE and UZK are the sensitive parameters (see Fig. 6). The Metamodel-based Sobol' method not only identifies the most sensitive parameters, but also quantifies the contribution ratio of each parameter. For the Metamodel-based Sobol' method, we conclude that parameters LZPK, PCTIM, ADIMP, UZK, LZSK and PFREE are the sensitive parameters (see Fig. 7). By comparing Figs. 6 and 7, we note that the SA results of the MOAT and Metamodel based Sobol' methods are consistent, both identifying LZPK, PCTIM, ADIMP, UZK, LZSK and PFREE as the most sensitive parameters, with the latter showing

Table 4
Surrogate modeling methods available in UQ-PyL.

Surrogate modeling method	Source
Polynomial	Fen et al., 2009
Generalized Linear Model	Hastie and Tibshirani, 1990
Regression Tree	Breiman et al., 1984
Random Forest	Breiman, 2001
MARS	Friedman, 1991
Nearest Neighbors	Bentley, 1975
Support Vector Machine	Zhang et al., 2009
Gaussian Process	Rasmussen and Williams, 2006

Table 3
SA methods available in UQ-PyL.

SA type	SA method	Source
Derivative-based	MOAT	Morris 1991 Campolongo et al., 2007
	Derivative-based Global Sensitivity Measure (DGSM)	Sobol' and Kucherenko, 2009
Variance-based	Sobol'	Sobol' 2001, Saltelli 2002, Saltelli et al., 2010
	Fourier Amplitude Sensitivity Test (FAST)	Cukier et al., 1973, Saltelli et al., 1999
	Metamodel-based Sobol'	Gan et al., 2014
Regression-based	Correlation analysis	Spearman, 1904
	Gaussian process screening	Gibbs and Mackay, 1997
	MARS screening	Friedman, 1991
	Delta Moment-Independent Measure	Borgonovo 2007, Plischke et al., 2013

Table 5
Parameter optimization methods available in UQ-PyL.

Algorithm type	Optimization method	Source
Deterministic	Shuffled Complex Evolution	Duan et al., 1992
	Simulated Annealing	Kirkpatrick et al., 1983
	Dynamically Dimensional Search	Tolson and Shoemaker, 2007
	Adaptive Surrogate Modeling based Optimization	Wang et al., 2014
Bayesian	Monte Carlo Markov Chain	Geyer, 1992; Neal, 1993

Table 6
Parameters of SAC-SMA model.

No.	Parameter	Description	Range
1	UZTWM	Upper zone tension water maximum storage (mm)	[10, 300]
2	UZFWM	Upper zone free water maximum storage (mm)	[5, 150]
3	UZK	Upper zone free water lateral drainage rate (day^{-1})	[0.1, 0.75]
4	PCTIM	Impervious fraction of the watershed area (decimal fraction)	[0, 0.1]
5	ADIMP	Additional impervious area (decimal fraction)	[0, 0.2]
6	ZPERC	Maximum percolation rate (dimensionless)	[5, 350]
7	REXP	Exponent of the percolation equation (dimensionless)	[1, 5]
8	LZTWM	Lower zone tension water maximum storage (mm)	[10, 500]
9	LZFSM	Lower zone supplemental free water maximum storage (mm)	[5, 400]
10	LZFPM	Lower zone primary free water maximum storage (mm)	[10, 1000]
11	LZSK	Lower zone supplemental free water lateral drainage rate (day^{-1})	[0.01, 0.35]
12	LZPK	Lower zone primary free water lateral drainage rate (day^{-1})	[0.001, 0.05]
13	PFREE	Fraction of water percolating from upper zone directly to lower zone free water (decimal fraction)	[0.0, 0.9]
14	RIVA	Riverside vegetation area (decimal fraction)	0.3
15	SIDE	Ration of deep recharge to channel base flow (dimensionless)	0
16	RSERV	Fraction of lower zone free water not transferrable to lower zone tension water (decimal fraction)	0

the exact sensitivity indexes as well (see Fig. 7). The sensitivity results from those figures are basically consistent, except UZK, whose sensitivity ranks in the two figures changed places. This difference is sensible as it is due to sampling uncertainty.

4.4. The demonstration of surrogate modeling methods

The objective of this section is to demonstrate the different surrogate modeling methods. Three algorithms available from the

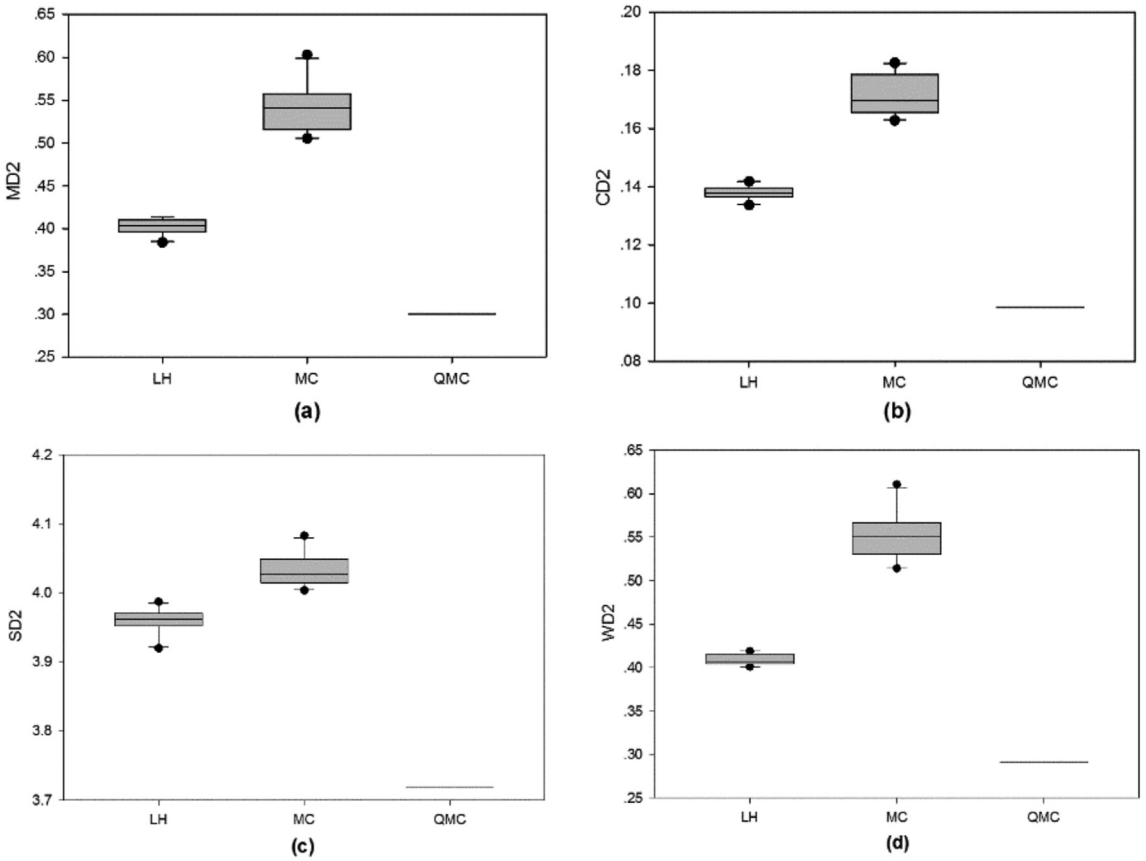

Fig. 5. Comparison of four metric values on three DoE methods, (a) is MD2 metric, (b) is CD2 metric, (c) is SD2 metric and (d) is WD2 metric.

Table 7
Moments results of SAC-SMA model's objective function value.

Mean	Variance	Skewness	Kurtosis
0.6697	0.0506	3.8326	38.2259

Table 8
Confidence interval of SAC-SMA model's mean of objective function value.

0.30%	5.00%	32.60%	50.00%	67.40%	95.00%	99.70%
0.640	0.650	0.660	0.670	0.680	0.689	0.700

UQ-PyL are chosen for this purpose: Regression tree (Tree), Support Vector Machine (SVM) and Gaussian Process (GP). Tree is a non-parametric supervised learning method used to predict a response from several inputs. It can be conveniently used for Bayesian supervised learning, such as regression. SVM is a learning machine implementing the structural risk minimization inductive principle to obtain good generalization on a limited number of learning patterns (Basak et al., 2007). GP is an interpolating regression method that uses a basis function with tuned parameters to represent the original model (Jones, 2001). To check the performance of the surrogate modeling methods, we used the K-fold cross-validation (Picard and Cook, 1984) method which works as follows. The entire sample set is divided into K subsets. Given a specific surrogate construction method, a response surface is created by using only K-1 subsets of sample points. The difference between the true response surface and the response surface built on the K-1 subsets of samples is computed. This procedure iterates K times to obtain the average predictive error. If the error is within specified tolerance, the surrogate model is regarded as acceptable.

Fig. 8 compares the predictive errors of different surrogate model methods. From this figure, we note that GP performs the best among these three methods. For subsequent analyses, we use the surrogate model created by GP for further analyses.

4.5. The demonstration of parameter optimization methods

The last step in UQ process is to perform parameter

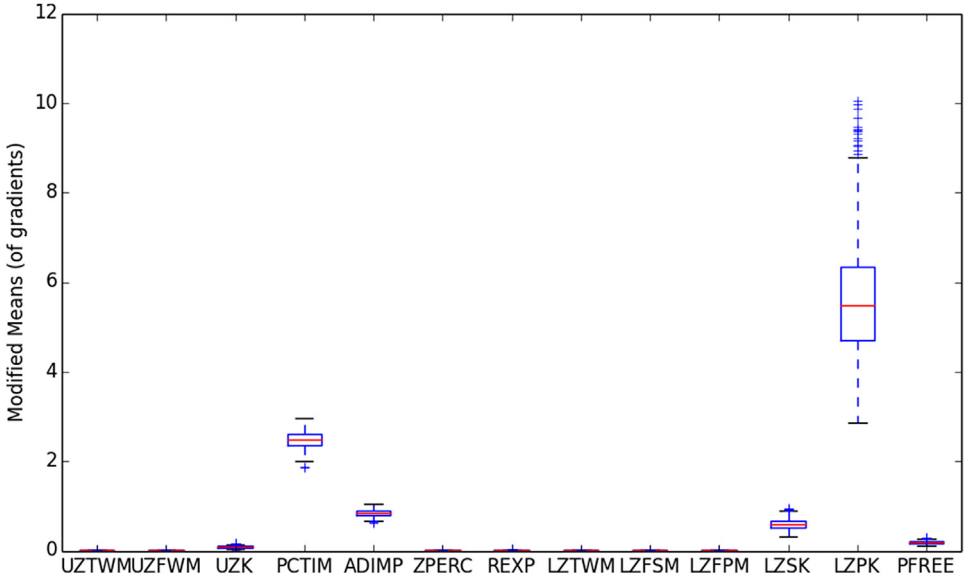

Fig. 6. Sensitivity analysis result of MOAT overall effect on SAC-SMA model.

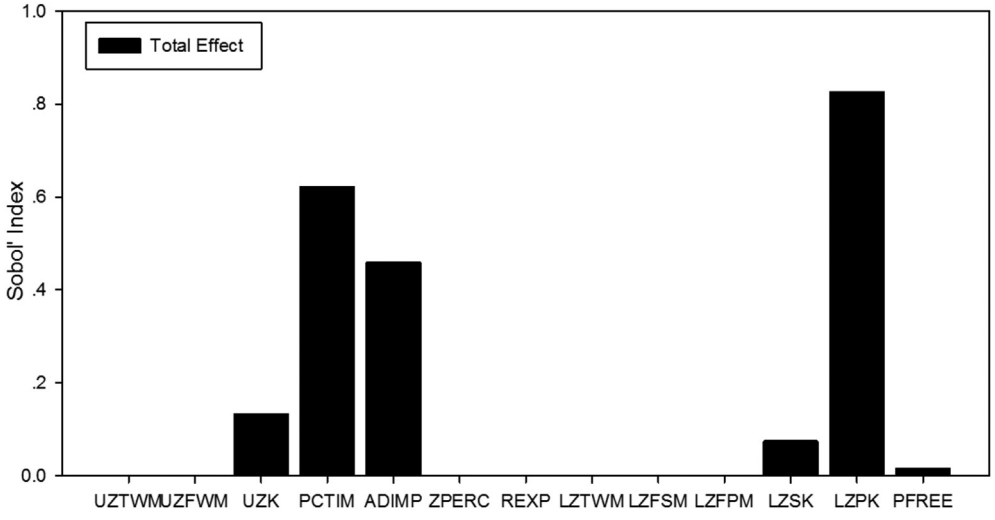

Fig. 7. Sensitivity analysis results of Metamodel based Sobol' total effect on SAC-SMA model.

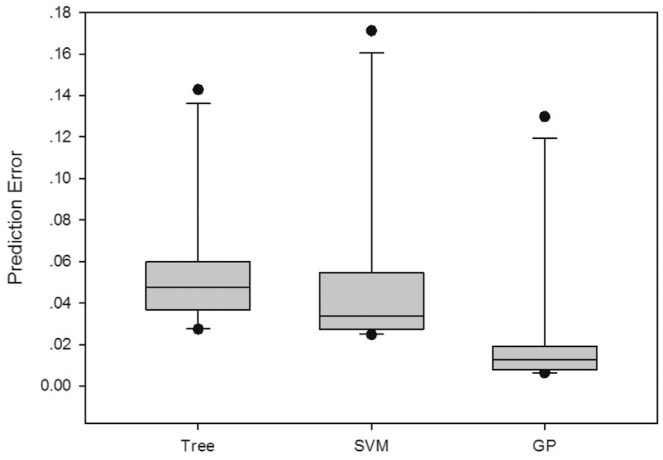

Fig. 8. Box plot of predictive errors of different surrogate model methods.

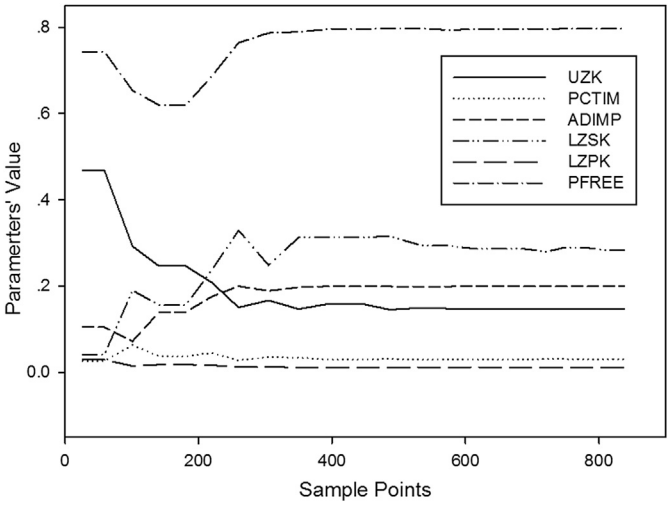

Fig. 9. Parameter optimization results of SCE-UA on 6-parameter SAC-SMA.

optimization. After sensitivity analysis, we conclude that there are six sensitive parameters (LZPK, PCTIM, ADIMP, UZK, LZSK and PFREE) in SAC-SMA. In this section, we will only optimize these six parameters. The other parameters are set to their respective default values. There are several optimization methods available in UQ-PyL. See Table 5 for the listing. We choose two typical parameter optimization algorithms – Shuffled Complex Evolution (SCE-UA) (Duan et al., 1992, 1994) and Monte Carlo Markov Chain (MCMC) as examples. SCE-UA is an evolution based directed global optimization algorithm which is designed to find a unique set of global optimal parameter set. Many studies have demonstrated that SCE-UA is an effective and efficient method for parameter calibration (Gan and Biftu, 1996; Hogue et al., 2000; Moreno et al., 2012). Fig. 9 shows the parameter traces during optimization search using the SCE-UA method. We note that, after about 800 sample points, the optimization search converges to its optimal solution, with an objective function value of 0.945. Different from SCE-UA, MCMC assumes that there is no unique set of optimal parameters. Instead, MCMC searches for the posterior distribution of the model parameters which maximize the likelihood of model simulation matching corresponding observation. Fig. 10 shows the marginal posterior distributions of the six optimized parameters. Five of the six parameters are clearly identifiable, with clear peaks in distribution, except LZSK, which seems to non-identifiable.

5. Conclusion

In this paper, we introduced a newly developed UQ software platform – UQ-PyL. It contains a wide variety of UQ algorithms, including DoE, statistical analysis, SA, surrogate modeling and global optimization. UQ-PyL uses a consistent, task-oriented framework, thus enables easy experiments with different methods for a given problem. It is designed in a way that it is easy to add new algorithms to its library. Further, UQ-PyL is implemented in a high-level Python language. The GUI equipped in the software allows an easy setup and execution of different UQ analyses. It runs across different platforms including Windows, Linux and MacOS.

The UQ-PyL software provides a very handy tool for UQ analyses of complex computer models. It is easy to be coupled with user

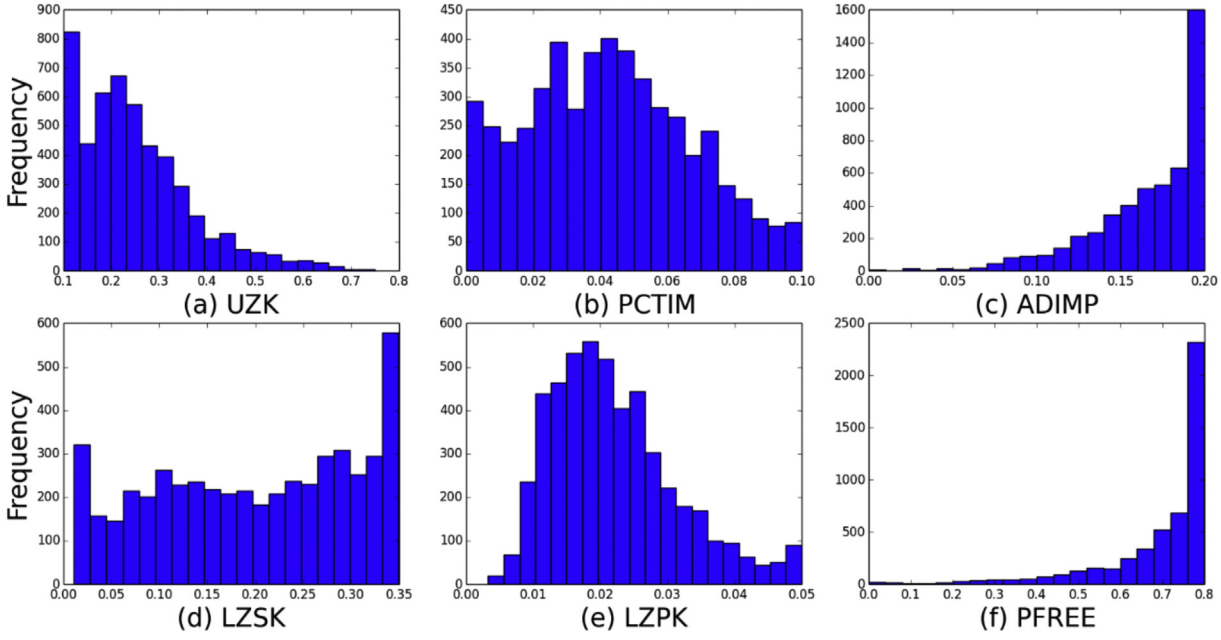

Fig. 10. Parameter optimization results of MCMC on 6-parameter SAC-SMA.

defined external dynamical simulation models. The user needs to provide only three pieces of information, including model executable file, model configuration file and model simulation outputs of interest, along with the corresponding observations and the error function to use UQ-PyL. We demonstrate various functions of UQ-PyL through a simple hydrological model SAC-SMA and the results obtained by different UQ algorithms from UQ-PyL are reasonable. The examples shown in the paper were designed to facilitate users explore the capabilities of UQ-PyL fully.

Acknowledgment

This research is supported by the Ministry of Science and Technology of the People's Republic of China National Science and Technology Support Program (No. 2013BAB05B04 and No. 41375139).

References

Adams, B.M., Bauman, L.E., Bohnhoff, W.J., Dalbey, K.R., Ebeida, M.S., Eddy, J.P., Eldred, M.S., Hough, P.D., Hu, K.T., Jakeman, J.D., Swiler, L.P., Vigil, D.M., December 2009. DAKOTA, a Multilevel Parallel Object-oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.4 User's Manual. Sandia Technical Report SAND2010–2183. Updated April 2013.

Andrianov, G., Burriel, S., Cambier, S., Dutfoy, A., Dutka-Malen, I., de Rocquigny, E., Sudret, B., Benjamin, P., Lebrun, R., Mangeant, F., Pendola, M., 2007. OpenTURNS, an open source initiative to treat uncertainties, Risks'N statistics in a structured industrial approach. In: Proceedings of the ESREL'2007 Safety and Reliability Conference, Stavenger: Norway.

Basak, D., Pal, S., Patranabis, D., 2007. Support vector regression. Neural Inf. Process Lett. Rev. 11 (10).

Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. Commun. ACM 18 (9), 509–517.

Borgonovo, E., 2007. A new uncertainty importance measure. Reliab. Eng. Syst. Saf. 92 (6), 771–784.

Box, G., Behnken, D., 1960. Some new three level designs for the study of quantitative variables. Technometrics 2, 455–475.

Box, G.E., Hunter, J.S., Hunter, W.G., 2005. Statistics for Experimenters: Design, Innovation, and Discovery, second ed. Wiley, ISBN 0-471-71813-0.

Box, G.E.P., Draper, N.R., 1987. Empirical Model-building and Response Surfaces. Wiley.

Brazil, L.E., 1988. Multilevel Calibration Strategy for Complex Hydrologic Simulation Models. Ph.D. Thesis. Department of Civil Engineering, Colorado State University, Fort Collins, CO.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Wadsworth, Belmone, California.

Breiman, L., 2001. Random forests. Mach. Learn 45 (1), 5–32. http://dx.doi.org/10.1023/A.1010933404324.

Burnash, R.J.C., 1995. The NWS river forecast system: catchment modeling. In: Singh, V.P. (Ed.), Computer Models of Watershed Hydrology. Water Resources Publications, Highlands Ranch, Colorado, pp. 311–366.

Burnash, R.J.C., Ferral, R.L., McGuire, R.A., 1973. A Generalized Streamflow Simulation System: Conceptual Modeling for Digital Computers. US Department of Commerce, National Weather Service, Sacramento, CA.

Campolongo, F., Cariboni, J., Saltelli, A., 2007. An effective screening design for sensitivity analysis of large models. Environ. Model. Softw. 22 (10), 1509–1518.

Casella, G., Berger, R.L., 2002. Statistical Inference, second ed. Duxbury, Pacific Grove, ISBN 0-534-24312-6.

Cox, D.R., Hinkley, D.V., 1974. Theoretical Statistics. Chapman & Hall.

Cukier, R.I., Fortuin, C.M., Shuler, K.E., Petschek, A.G., Schaibly, J.H., 1973. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I Theory. J. Chem. Phys. 59 (8), 3873–3878.

Doherty, J., 2004. PEST, Model-independent Parameter Estimation, User Manual, fifth ed. Watermark Numerical Computing, Australia.

Du, J., 2007. Uncertainty and Ensemble Forecast. In: Science & Technology Infusion Lecture Series.

Duan, Q.Y., Sorooshian, S., Gupta, V., 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resour. Res. 28 (4), 1015–1031.

Duan, Q.Y., Sorooshian, S., Gupta, V.K., 1994. Optimal use of the SCE-UA global optimization method for calibrating watershed models. J. Hydrol. 158 (3–4), 265–284.

Duan, Q., Schaake, J., Andreassian, V., Franks, S., Goteti, G., Gupta, H.V., Gusev, Y.M., Habets, F., Hall, A., Hay, L., Hogue, T., Huang, M., Leavesley, G., Liang, X., Nasonova, O.N., Noilhan, J., Oudin, L., Sorooshian, S., Wagener, T., Wood, E.F., 2006. Model parameter estimation experiment (MOPEX): an overview of science strategy and major results from the second and third workshops. J. Hydrol. 320 (1–2), 3–17.

Easton, V.J., McColl, J.H., 1997. Statistics Glossary v1.1. The STEPS Project.

Fen, C.S., Chan, C.C., Cheng, H.C., 2009. Assessing a response surface-based optimization approach for soil vapor extraction system design. J. Water Resour. Plan. Manag. 135 (3), 198–207.

Friedman, J.H., 1991. Multivariate adaptive regression splines. Ann. Statistics 19 (1), 1–67.

Gan, T.Y., Biftu, G.F., 1996. Automatic calibration of conceptual rainfall-runoff models: optimization algorithms, catchment conditions, and model structure. Water Resour. Res. 32 (12), 3513–3524.

Gan, Y.J., Duan, Q.Y., Gong, W., Tong, C., Sun, Y.W., Chu, W., Ye, A.Z., Miao, C.Y., Di, Z.H., 2014. A comprehensive evaluation of various sensitivity analysis methods: a case study with a hydrological model. Environ. Model. Softw. 51, 269–285.

Geyer, C.J., 1992. Practical markov chain monte carlo. Stat. Sci. 473–483.

Gibbs, M., MacKay, D.J.C., 1997. Efficient Implementation of Gaussian Processes (Unpublished manuscript).

Gong, W., Duan, Q.Y., et al., 2015. An inter-comparison of sampling methods for large complex dynamic models. Submitted to J. Environ. Inf.. http://dx.doi.org/10.3808/jei.201500310.

Hastie, T.J., Tibshirani, R.J., 1990. Generalized Additive Models. Chapman & Hall/CRC, ISBN 978-0-412-34390-2.

Hogue, T.S., Sorooshian, S., Gupta, H., Holz, A., Braatz, D., 2000. A multistep automatic calibration scheme for river forecasting models. J. Hydrometeorol. 1 (6), 524–542.

Houtekamer, P.L., Mitchell, H.L., 1998. Data assimilation using an ensemble Kalman Filter technique. Mon. Weather Rev. 126, 796–811.

Iman, R.L., Helton, J.C., Campbell, J.E., 1981. An approach to sensitivity analysis of computer models, part 1. Introd. input Var. Sel. Prelim. Var. Assess. 13 (3), 174–183.

Ji, D., Wang, L., Feng, J., Wu, Q., Cheng, H., Zhang, Q., Yang, J., Dong, W., Dai, Y., Gong, D., Zhang, R.H., Wang, X., Liu, J., Moore, J.C., Chen, D., Zhou, M., 2014. Description and basic evaluation of Beijing Normal University Earth system model (BNU-ESM) version 1. Geosci. Model Dev. 7 (5), 2039–2064.

Jones, D., 2001. A taxonomy of global optimization methods based on response surfaces. J. Glob. Optim. 21, 345–383.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680.

Krykova, I., 2003. Evaluating of path-dependent securities with low discrepancy methods. In: Financial Mathematics. Worcester Polytechnic Institute.

Li, J.D., Duan, Q.Y., Gong, W., Ye, A., Dai, Y., Miao, C.Y., Di, Z., Tong, C., Sun, Y., 2013. Assessing parameter importance of the Common Land Model based on qualitative and quantitative sensitivity analysis. Hydrol. Earth Syst. Sci. 17, 3279–3293. http://dx.doi.org/10.5194/hess-17-3279-2013.

Maître, O.P.Le, Knio, O.M., 2010. Spectral Methods for Uncertainty Quantification, with Applications to Computational Fluid Dynamics. Springer.

Marelli, S., Sudret, B., 2014. UQLab: a framework for uncertainty quantification in Matlab. Vulnerability, Uncertain. Risk 2554–2563. http://dx.doi.org/10.1061/9780784413609.257.

Meteopolis, N., Ulam, S., 1949. The Monte Carlo method. J. Am. Stat. Assoc. 44 (247), 335–341.

Mckay, M.D., Beckman, R.J., Conover, W.J., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21 (2), 239–245. http://dx.doi.org/10.2307/1268522.

Moreno, H.A., Vivoni, E.R., Gochis, D.J., 2012. Utility of quantitative precipitation estimates for high resolution hydrologic forecasts in mountain watersheds of the Colorado front range. J. Hydrol. 438–439, 66–83. http://dx.doi.org/10.1016/j.jhydrol.2012.03.019.

Morris, M.D., 1991. Factorial sampling plans for preliminary computational experiments. Technometrics 33 (2), 161–174.

Myers, R.H., 1971. Response Surface Methodology. Allyn and Bacon, Inc., Boston.

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

Niederreiter, H., 1992. Random number generation and quasi-Monte Carlo methods. In: CBMS-NSF, vol. 63. SIAM, Philadelphia.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Picard, R.R., Cook, R.D., 1984. Cross-validation of regression models. J. Am. Stat. Assoc. 79 (387), 575–583.

Plackett, R.L., Burman, J.P., 1946. The design of optimum multifactorial experiments. Biometrika 33 (4), 305–325.

Plischke, E., Borgonovo, E., Smith, C.L., 2013. Global sensitivity measures from given data. Eur. J. Operational Res. 226 (3), 536–550.

Poeter, E.P., Hill, M.C., Banta, E.R., Mehl, S., Christensen, S., 2005. UCODE_2005 and Six Other Computer Codes for Universal Sensitivity Analysis, Calibration, and Uncertainty Evaluation. http://typhoon.mines.edu/freeware/ucode/.

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. MIT Press, ISBN 026218253X.

Saltelli, A., 2002. Making best use of model evaluations to compute sensitivity indices. Comput. Phys. Commun. 145 (2), 280–297.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., 2008. Global Sensitivity Analysis: the Primer. John Wiley & Sons, Ltd, Chichester, West Sussex.

Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M., 2004. Sensitivity Analysis in Practise – a Guide to Assessing Scientific Models. John Wiley & Sons, Ltd,

Chichester, West Sussex.
Saltelli, A., Tarantola, S., Chan, K.P.S., 1999. A quantitative model-independent method for global sensitivity analysis of model output. Technometrics 41 (1), 39–56.
Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S., 2010. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Comput. Phys. Commun. 181 (2), 259–270.
Schervish, M., 1996. Theory of Statistics. Springer, ISBN 0-387-94546-6.
Shin, M.-J., Guillaume, J.H.A., Croke, B.F.W., Jakeman, A.J., 2015. A review of foundational methods for checking the structural identifiability of models: results for rainfall-runoff. J. Hydrol. 520, 1–16. http://dx.doi.org/10.1016/j.jhydrol.2014.11.040.
Shin, M.-J., Guillaume, J.H.A., Croke, B.F.W., Jakeman, A.J., 2013. Addressing ten questions about conceptual rainfall-runoff models with global sensitivity analyses in R. J. Hydrol. 503, 135–152. http://dx.doi.org/10.1016/j.jhydrol.2013.08.047.
Sobol', I.M., 1967. On the distribution of points in a cube and the approximate evaluation of integrals. Zhurnal Vychislitel'noi Mat. i Mat. Fiz. 7 (4), 784–802.
Sobol', I.M., 2001. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Math. Comput. Simul. 55 (1–3), 271–280.
Sobol', I.M., Kucherenko, S., 2009. Derivative based global sensitivity measures and their link with global sensitivity indices. Math. Comput. Simul. 79 (10), 3009–3017.
Spearman, C., 1904. The proof and measurement of association between two things. Am. J. Psychol. 15 (1), 72–101.
Tolson, B.A., Shoemaker, C.A., 2007. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. Water Resour. Res. 43 (1).
Tong, C., 2005. PSUADE User's Manual. Lawrence Livermore National Laboratory (LLNL), Livermore, CA.
Wang, C., Duan, Q.Y., Gong, W., Ye, A.Z., Di, Z.H., Miao, C.Y., 2014. An evaluation of adaptive surrogate modeling based optimization with two benchmark problems. Environ. Model. Softw. 60, 167–179.
Wu, C.F.J., Hamada, M., 2009. Experiments: Planning, Analysis, and Optimization, second ed. Wiley, New York.
Ye, K.Q., Li, W., Sudjianto, A., 2000. Algorithmic construction of optimal symmetric latin hypercube designs. J. Stat. Plan. Inference 90, 145–159.
Zhang, X.S., Srinivasan, R., Van Liew, M., 2009. Approximating SWAT model using artificial neural network and support vector machine. J. Am. Water Resour. Assoc. 45 (2), 460–474.
Ziehn, T., Tomlin, A.S., 2009. GUI-HDMR - a software tool for global sensitivity analysis of complex models. Environ. Model. Softw. 24 (7), 775–785. http://dx.doi.org/10.1016/j.envsoft.2008.12.002.