Environmental Modelling & Software 51 (2014) 269–285

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

A comprehensive evaluation of various sensitivity analysis methods: A case study with a hydrological model☆

Yanjun Gan [a], Qingyun Duan [a,*], Wei Gong [a], Charles Tong [b], Yunwei Sun [c], Wei Chu [d], Aizhong Ye [a], Chiyuan Miao [a], Zhenhua Di [a]

[a] State Key Laboratory of Earth Surface Processes and Resource Ecology, College of Global Change and Earth System Science, Beijing Normal University, Beijing 100875, China
[b] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551-0808, USA
[c] Atmosphere, Earth and Energy Division, Lawrence Livermore National Laboratory, Livermore, CA 94551-0808, USA
[d] Department of Civil and Environmental Engineering, University of California, Irvine, CA 92617, USA

ARTICLE INFO

Article history:
Received 17 May 2013
Received in revised form
30 September 2013
Accepted 30 September 2013
Available online 12 November 2013

Keywords:
Uncertainty quantification
Sensitivity analysis
Parameter screening
Space-filling sampling
PSUADE

ABSTRACT

Sensitivity analysis (SA) is a commonly used approach for identifying important parameters that dominate model behaviors. We use a newly developed software package, a Problem Solving environment for Uncertainty Analysis and Design Exploration (PSUADE), to evaluate the effectiveness and efficiency of ten widely used SA methods, including seven qualitative and three quantitative ones. All SA methods are tested using a variety of sampling techniques to screen out the most sensitive (i.e., important) parameters from the insensitive ones. The Sacramento Soil Moisture Accounting (SAC-SMA) model, which has thirteen tunable parameters, is used for illustration. The South Branch Potomac River basin near Springfield, West Virginia in the U.S. is chosen as the study area. The key findings from this study are: (1) For qualitative SA methods, Correlation Analysis (CA), Regression Analysis (RA), and Gaussian Process (GP) screening methods are shown to be not effective in this example. Morris One-At-a-Time (MOAT) screening is the most efficient, needing only 280 samples to identify the most important parameters, but it is the least robust method. Multivariate Adaptive Regression Splines (MARS), Delta Test (DT) and Sum-Of-Trees (SOT) screening methods need about 400–600 samples for the same purpose. Monte Carlo (MC), Orthogonal Array (OA) and Orthogonal Array based Latin Hypercube (OALH) are appropriate sampling techniques for them; (2) For quantitative SA methods, at least 2777 samples are needed for Fourier Amplitude Sensitivity Test (FAST) to identity parameter main effect. McKay method needs about 360 samples to evaluate the main effect, more than 1000 samples to assess the two-way interaction effect. OALH and LP_τ (LPTAU) sampling techniques are more appropriate for McKay method. For the Sobol' method, the minimum samples needed are 1050 to compute the first-order and total sensitivity indices correctly. These comparisons show that qualitative SA methods are more efficient but less accurate and robust than quantitative ones.

Software availability

Name of software: PSUADE
Developer: Charles Tong
Programming language: C++
Availability: https://computation.llnl.gov/casc/uncertainty_quantification/
Cost: Free for non-commercial academic research

1. Introduction

Computer-based system models have become indispensable in many fields of science and engineering, from finance to life sciences, from quantum physics to earth sciences and environmental engineering. Parameters of these models exert great influence on models' performance. Some of the parameters may be observed or measured, e.g., the physical dimensions of an object or the geomorphological features of a watershed such as slope, area size and elevation. But there are many parameters that are not directly observable, at least not at the scale of modeling units. For example, parameters commonly used in hydrologic models, such as saturated soil hydraulic conductivity or saturated soil matric potential, may be observable at a point scale, but not over a large area. In

* Corresponding author. Tel.: +86 10 5880 4191; fax: +86 10 5880 2165.
E-mail address: qyduan@bnu.edu.cn (Q. Duan).

1364-8152/$ – see front matter © 2013 The Authors. Published by Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.envsoft.2013.09.031

this case, "effective" values must be estimated so mathematical equations established at a point scale can be extended to an areal scale (Blöschl and Sivapalan, 1995). There is a class of models known as conceptual models whose parameters are generally non-observable and are only related to physical properties indirectly. For example, the parameters in many conceptual rainfall-runoff (CRR) models are not observable and must be calibrated so model simulations closely match observations (Duan et al., 1992).

How to specify system model parameters properly is not a trivial issue (Sorooshian and Gupta, 1983; Duan et al., 1992, 2006; Kavetski et al., 2003). The combined effect of several factors, including errors in observational data, choices of calibration methods and criterias, and model formulation errors, makes parameter estimation being a difficult task. This difficulty is further compounded by over-parameterization problems as today's models are getting increasingly complex in a trend to include more and more sub-physics, but the calibration of these models is still done with rather limited data (Jakeman and Hornberger, 1993; Renard et al., 2010; Clark et al., 2011). Over-parameterization, along with parameter interactions (due to high nonlinearity of model equations), causes model parameters to be not uniquely identifiable. Beven (2006) termed this phenomenon as equifinality, i.e., different parameter sets would result in the same or similar model performance measures. Another potential cause for equifinality may be due to a phenomenon known as "numerical daemon" by Kavetski and Clark (2010). One possible way to mitigate over-parameterization/non-identifiability is reducing the number of parameters to a small number that can be sufficiently calibrated with limited data.

To discern which parameters have the most influence over model performance and to identify what are the most appropriate parameter values, we need to find a way to screen out sensitive parameters and quantitatively evaluate the influence of each parameter on model performance. Sensitivity analysis (SA) has been used by many people for this purpose (Liu et al., 2004; van Griensven et al., 2006; Campolongo et al., 2007; Borgonovo et al., 2012). SA can identify parameters of which a reduction in uncertainty specification will have the most significant impact on improving model performance measures. Thus, if some non-influential parameters can be identified and fixed reasonably at given values over their ranges, the computational cost may decrease without reducing model performance.

There are many different SA approaches. Overall, they can be categorized into two groups: local SA and global SA. The local SA explores the changes of model response by varying one parameter while keeping other parameters constant. The simplest and most common approach is differential SA (DSA), which uses partial derivatives or finite differences of parameters at a fixed parameter location as the measure of parametric sensitivity. Though simple and intuitive, DSA measures only local sensitivity whose value is obviously location dependent. On the other hand, the global SA examines the changes of model response by varying all parameters at the same time. Generalized SA (GSA) method is one of the global SA methods that are designed to overcome the limitations of local SA methods. A version of GSA method, as implemented in Hornberger and Spear (1981), first creates a large number of random parameter sets using the Monte Carlo (MC) (Meteopolis and Ulam, 1949) sampling technique. It then breaks the random parameter sets into behavioral and non-behavioral sets based on a pre-specified threshold for acceptance of model behavior. The frequency density distributions of model performance measures along each parameter axis in the behavioral sub-set are used as indicators of parametric sensitivities. GSA forms the basis for the Generalized Likelihood Uncertainty Estimation (GLUE) method developed by Beven and Binley (1992). GSA is simple to implement and can work with different pseudo-likelihood (i.e., goodness of fit) measures (Beven, 2004), but it is computationally inefficient.

Global SA approaches based on design of experiment (DOE) have gained popularity recently because they offer global sensitivity measures while maintaining computational efficiency. A typical DOE-based SA method involves two steps: first, generating a sample set of parameters within the feasible parameter spaces using a chosen design; and then, obtaining a quantitative attribution of model output variation due to the variation of different parameters. There are many sampling techniques, such as MC, Latin Hypercube (LH) (McKay et al., 1979), Orthogonal Array (OA) (Owen, 1992) and Orthogonal Array based Latin Hypercube (OALH) (Tang, 1993), which are commonly used for DOE-based SA. Some DOE-based SA methods, such as Morris One-At-a-Time (MOAT) (Morris, 1991), Fourier Amplitude Sensitivity Test (FAST) (Cukier et al., 1973), and extended Sobol' method (Saltelli, 2002), require special sampling techniques. More recently, along with the development of response surface methods (RSM), SA based on RSM makes it cheaper for estimating parameter effects (Ratto et al., 2007; Shahsavani and Grimvall, 2011).

Saltelli et al. (2008) provided a comprehensive exposition of contemporarily available SA methods. Tong (2005) developed a software package, called a Problem Solving environment for Uncertainty Analysis and Design Exploration (PSUADE) and containing a wide array of different uncertainty quantification (UQ) methods, including many SA methods. PSUADE has been used successfully for many applications. Hsieh (2006) demonstrated the process of using PSUADE for UQ of the Steven Impact Test problem. Wemhoff and Hsieh (2007) used PSUADE to calibrate the Prout–Tompkins chemical kinetic model. Tong (2008) applied a variety of UQ techniques to the study of a two-dimensional soil-foundation structure-interaction system subjected to earthquake excitation using PSUADE. Tong and Graziani (2008) described a global SA methodology implemented in PSUADE that is specifically designed for general multi-physics application of large complex system models. Snow and Bajaj (2010) adopted the PSUADE for uncertainty analysis of a comprehensive electrostatic Micro-Electro Mechanical Systems (MEMS) switch model.

The aforementioned works have been focused on applying a subset of the UQ methods available within PSUADE. The purpose of this paper is to explore the effectiveness and efficiency of various SA methods in PSUDAE in identifying sensitive parameters of system models, and provide useful guidance on selecting appropriate SA procedures for other applications. We test all available SA methods with a very simple conceptual hydrologic model – Sacramento Soil Moisture Accounting (SAC-SMA) model (Burnash et al., 1973). The generality of the findings in this paper would need further works on more complex models and more catchments with different characteristics. This paper is organized as follows. Section 2 offers a brief description of the PSUADE software. Section 3 describes the model, data and experimental methods used in the study. Section 4 presents the results and discussion. And finally, we make some concluding remarks in Section 5.

2. The PSUADE software

PSUADE is a C++ based open-source software package developed to provide an integrated design and analysis environment for performing UQ for large complex system models. This software is available via https://computation.llnl.gov/casc/uncertainty_quantification/. The flow chart for implementing PSUADE for UQ is shown in Fig. 1. The three parts in bold italic are basic elements of PSUADE:

- The experimental design techniques (***Sample generator***)
- The simulator execution environment (***Driver***)
- The analysis toolset (***Analysis tool***)