

Short communication

A Matlab toolbox for Global Sensitivity Analysis



Francesca Pianosi*, Fanny Sarrazin, Thorsten Wagener

Department of Civil Engineering, University of Bristol, University Walk, BS81TR, Bristol, UK

ARTICLE INFO

Article history:

Received 14 October 2014

Received in revised form

8 April 2015

Accepted 10 April 2015

Available online 15 May 2015

Keywords:

Global Sensitivity Analysis

Matlab

Octave

Open-source software

ABSTRACT

Global Sensitivity Analysis (GSA) is increasingly used in the development and assessment of environmental models. Here we present a Matlab/Octave toolbox for the application of GSA, called SAFE (Sensitivity Analysis For Everybody). It implements several established GSA methods and allows for easily integrating others. All methods implemented in SAFE support the assessment of the robustness and convergence of sensitivity indices. Furthermore, SAFE includes numerous visualisation tools for the effective investigation and communication of GSA results. The toolbox is designed to make GSA accessible to non-specialist users, and to provide a fully commented code for more experienced users to complement their own tools. The documentation includes a set of workflow scripts with practical guidelines on how to apply GSA and how to use the toolbox. SAFE is open source and freely available for academic and non-commercial purpose. Ultimately, SAFE aims at contributing towards improving the diffusion and quality of GSA practice in the environmental modelling community.

© 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Global Sensitivity Analysis (GSA) is a term describing a set of mathematical techniques to investigate how the variation in the output of a numerical model can be attributed to variations of its inputs. GSA can be applied for multiple purposes, including: to apportion output uncertainty to the different sources of uncertainty of the model, e.g. unknown parameters, measurement errors in input forcing data, etc. and thus prioritise the efforts for uncertainty reduction; to investigate the relative influence of model parameters over the predictive accuracy and thus support model calibration, verification and simplification; to understand the dominant controls of a system (model) and to support model-based decision-making.

Many GSA methods have been proposed in the literature and their application and comparison in the environmental modelling domain has steadily increased in recent years (e.g. Tang et al. (2007); Pappenberger et al. (2008); Yang (2011)). GSA has been recognised as an essential tool for the development and assessment of environmental models (Saltelli et al., 2008). However, the use of formal GSA techniques is still rather limited in some domains. Moreover, reported applications often fail to adequately tackle some critical issues like, in the first place, a rigorous assessment of

the robustness of GSA results to the multiple and sometimes non-univocal choices that the user has to make throughout its application. Tools are needed to facilitate uptake of the most advanced GSA techniques also by non-specialist users, as well as to provide guidelines on GSA application and to promote good practice.

Freely available GSA tools include the repository of Matlab and Fortran functions maintained by the Joint Research Centre (JRC, 2014), the Sensitivity Analysis package for the R environment (Pujol et al., 2014), the GUI-HDMR Matlab package (Ziehn and Tomlin, 2009), the C++ based PSUADE software (Gan et al., 2014), and the Python Sensitivity Analysis Library SALib (Herman, 2014). In this paper we present a Matlab toolbox for the application of GSA, called SAFE (Sensitivity Analysis For Everybody), specifically designed to conform with several principles that reflect the authors' view on "good practice" in GSA, namely: (i) the application of multiple GSA methods as a means to complement and validate individual results; (ii) the assessment and revision of the user choices made when applying each GSA method, especially in relation to the robustness of the estimated sensitivity indices; and (iii) the use of effective visualisation tools (see Table 1 for more discussion).

The SAFE Toolbox has primarily been conceived to make GSA accessible to non-specialist users, that is, people with only a basic background in GSA and/or Matlab. At the same time, it is designed to enable more experienced users to easily understand, customise, and possibly further develop the code. The toolbox documentation is also organised to meet these dual goals. It comprises a technical

* Corresponding author.

E-mail address: francesca.pianosi@bristol.ac.uk (F. Pianosi).

Table 1

Good practice in GSA applications and how they are made possible in SAFE.

Applying multiple methods. The application of different GSA methods to the same problem is advisable for at least two reasons. Firstly, as methods differ in their ability to address specific questions (e.g. input ranking, screening, mapping, analysis of individual contributions or of interactions (Saltelli et al., 2008)), the insights provided by several methods can complement each other so that a more complete picture of the problem at hand is obtained. Secondly, since methods rely on different assumptions (e.g. linear/non-linear input–output relationship, skewed/non-skewed output distribution) whose degree of validity is sometimes not clearly defined, the application of multiple methods is a practical way to validate, reject or reinforce the conclusions of GSA. The SAFE Toolbox has a modular structure that (i) makes it possible to re-use the same set of simulations for several GSA methods thus allowing for a multi-method approach while avoiding extra computational costs associated with new model evaluations; (ii) facilitates the integration of new GSA methods that the user may want to use for further comparison.

Assessing and revising the choices made. The user has to make a number of choices throughout the application of GSA, starting with the choice of the GSA method itself, the choice of the size of the feasible input space of variation, the choice of the sampling strategy for Monte Carlo simulations, etc. Often these choices are non-univocal and involve some degree of subjectivity. It is therefore important to enable the user to assess the robustness of the GSA results with respect to the choices made. When using sensitivity indices to measure output sensitivity, a particularly important issue is to evaluate the robustness of the index estimates. By robust we mean here that the index estimate does not significantly change if computed over a different sample of model simulations. In the SAFE Toolbox, any implemented sensitivity index can be associated with confidence intervals derived by bootstrapping and convergence analysis. Both the robustness assessment and convergence analysis do not require extra model evaluations and therefore they can easily be performed without adding to the overall computing cost of GSA.

Visualising GSA results. Effective visualisation tools are key for a successful application of GSA. Throughout the analysis, visualisation can support the user in exploring the results, especially when dealing with many inputs, for instance by facilitating the identification of outliers or counterintuitive behaviour, or by visualizing temporal or spatial patterns in output sensitivity, etc. Secondly, visualisation can support the communication of GSA results and conclusions. The SAFE Toolbox includes several functions implementing visual GSA methods (e.g. dot plots, posterior input distributions) and tools to visualise results of quantitative GSA (e.g. indices and associated uncertainty bounds). Colour scales in the functions have been conceived to maximise clarity using the Colorbrewer software (Brewer, 2013). The user can also switch any plotting function to black and white scale, for instance when preparing figures for publication.

Table 2

Documentation available for the SAFE Toolbox.

Technical documentation. This is directly embedded in the code through: (i) a 'function help' with details on the function inputs, outputs, and calling syntax, and a short description of the underlying method (with references); (ii) comments throughout the code that explain the rationale and specific steps of the implementation (intended for more experienced users).

User documentation. This is given in the form of several 'workflow' scripts that show, through practical examples, how the functions can be put together to utilize the Toolbox. An example of what a workflow looks like is given in Fig. 4. Workflows embody the good practice, which, in the authors' opinion, should guide the application of GSA. They can be used as tutorials to learn how to apply a specific method using the SAFE Toolbox but also to learn about the steps to be undertaken in developing a robust GSA in general. Workflows provide the added practical advantage that they can be used as a starting point to easily write new scripts by changing only the specific lines of code that define the experimental set-up and user choices. For all the above reasons we believe that workflow scripts constitute an effective and user-friendly way to develop User documentation.

documentation, which is embedded in the code, and a user documentation that is given in the form of workflow scripts (see Table 2). This paper complements that documentation by providing an overview of the Toolbox structure/architecture.

The first release of the SAFE Toolbox includes the Elementary Effects Test (EET, or Morris method (Morris, 1991)), Regional Sensitivity Analysis (RSA, Spear and Hornberger (1980); Wagener and Kollat (2007)), Variance-Based Sensitivity Analysis (VBSA, or Sobol' method, e.g. Saltelli et al. (2008)), the Fourier Amplitude Sensitivity Test (FAST by Cukier et al. (1973)), Dynamic identifiability analysis (DYNIA by Wagener et al. (2003)) and a novel density-based sensitivity method (PAWN by Pianosi and Wagener (2015)). The Toolbox also offers a number of visual tools including scatter (dot) plots, parallel coordinate plot and the visual test for validation of screening proposed by Andres (1997). The Toolbox has been designed to facilitate the integration of new methods and therefore this first release is meant to be the starting point of an ongoing code development project by the authors.

The SAFE Toolbox is implemented in Matlab but is also compatible with the freely available GNU Octave environment (www.gnu.org/software/octave/) and it runs under any operating system (Windows, Linux and Mac OS X). A R-version of the Toolbox is also available. Moreover, as it will be further described in the next section, the Toolbox can be easily linked to simulation models that run outside the Matlab/Octave environment. The Toolbox is freely available from the authors for noncommercial research and educational uses.

2. Structure of the SAFE Toolbox

Fig. 1 shows how the SAFE Toolbox is organised into folders. To better understand the file structure used in these folders, it must be

highlighted that all GSA approaches can be described through three basic steps (see Fig. 2):

- (1) Sampling the inputs within their variability space.
- (2) Evaluating the model against the sampled input combinations.
- (3) Post-processing the input/output samples to compute sensitivity indices.

Assuming that the simulation model of interest has already been implemented in a numerical programme, the application of a specific GSA method requires a set of functions to perform the first step (sampling) and the third step (post-processing). However, while the post-processing functions are tailored to each specific GSA approach, the same generic sampling function (for instance Latin Hypercube Sampling) can often be applied across different methods. Similarly, some visualisation tools can be used to visualise sensitivity indices estimated according to different methods (for instance the convergence plot shown in Fig. 3c can be used independently of the definition of the sensitivity index) or to provide additional insights to complement the GSA (an example is the widely used Parallel Coordinate Plot shown in Fig. 3d). Therefore, two types of folders in the SAFE Toolbox can be distinguished:

- shared folders (*sampling*, *util* and *visualisation*) that contain functions for sampling, visualisation, and other utilities, which might be used across different GSA methods;
- tailored folders (e.g. *EET*, *RSA* and *VBSA*) that contain the functions to compute sensitivity indices according to a specific method (e.g. the Elementary Effects Test, Regional Sensitivity Analysis, Variance-Based Sensitivity Analysis) and to visualise them in a method-specific fashion (for instance the elementary effects plot shown in Fig. 3a).

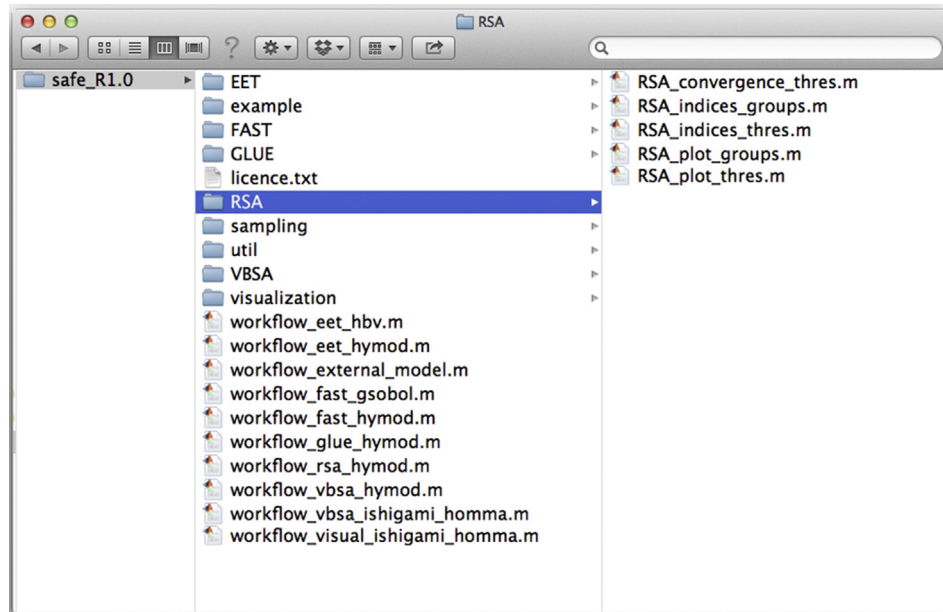


Fig. 1. Organization of the SAFE Toolbox.

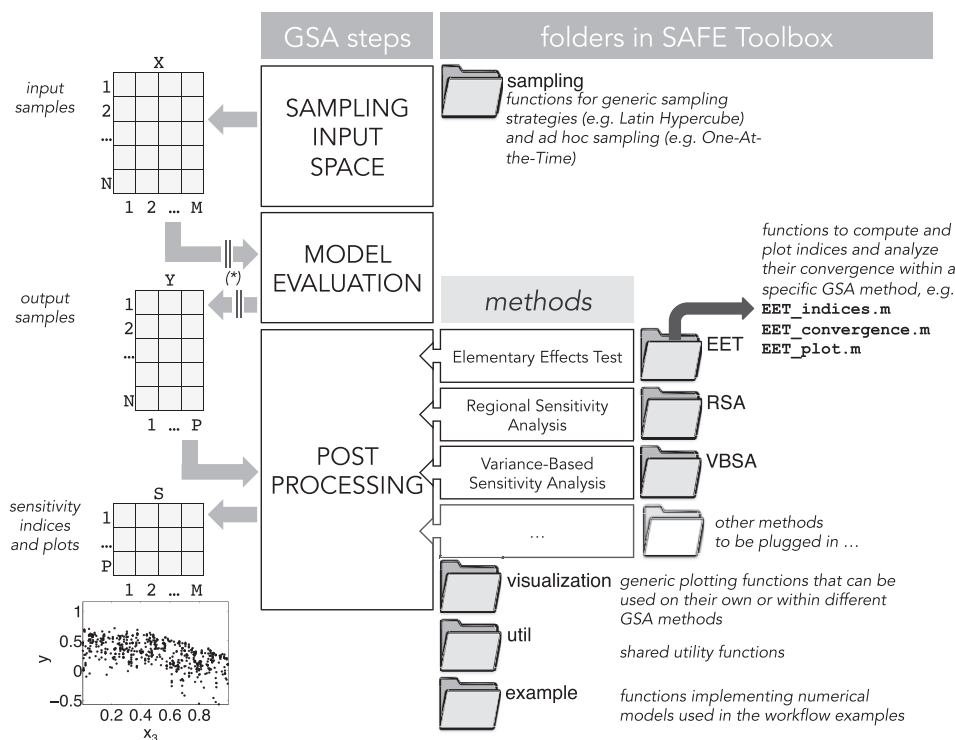


Fig. 2. The three basic steps of GSA and corresponding folders in the SAFE Toolbox (see Fig. 1). On left hand side of this Figure, the variables that each step takes as input and/or delivers as output: a matrix X of N randomly sampled input combinations (each made up of M components, M being the number of model inputs subject to GSA); a matrix Y of output samples (that can have $P > 1$ columns when evaluating the sensitivity of multiple model outputs); a matrix S of sensitivity indices. The asterisk indicates where variables may be exported/imported from/into Matlab to another computing environment.

This modular structure provides a number of advantages.

- It makes it easy to plug-in new code. For instance, new sampling methods can be included in the code by simply adding new functions to the *sampling* folder. The only requirement is that they produce an input sample matrix X in the format required by the post-processing functions (see Fig. 2). New GSA methods can also be easily integrated in the Toolbox. The

implemented functions will be grouped into a new folder, respecting the naming convention adopted in the Toolbox (i.e. `[methodname]_indices.m` for the function that computes the sensitivity indices, `[methodname]_plot.m` for the one that plots the indices, etc., see again Fig. 2 for an example). Again the only requirement for the integration is that all the post-processing functions have the sample matrices X and Y as input arguments.

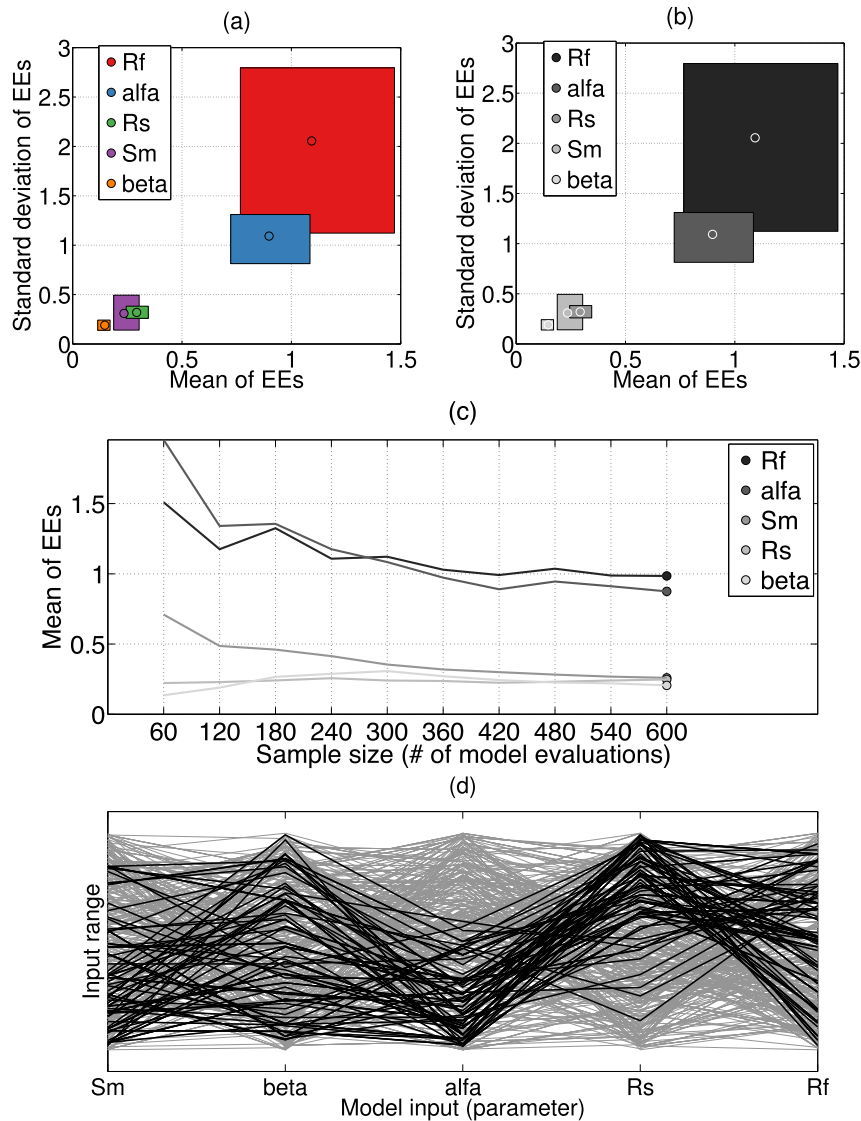


Fig. 3. Examples of visualisation tools implemented in the SAFE Toolbox (inputs of GSA are the 5 parameters [Rf, alfa, Rs, Sm, beta] of the rainfall-runoff Hymod model; the output is the Nash-Sutcliffe Efficiency NSE): (a) average of Elementary Effects against their standard deviation, with confidence bounds from bootstrapping; (b) same as before but in black and white (printer-friendly version); (c) convergence plot to analyse variations of the sensitivity index with the sample size (or required number of model evaluations); (d) parallel coordinate plot (in black, simulations where NSE>0.5).

- It makes it easy to use portions of the code only. For instance, if a dataset of input/output samples generated for a given model is already available (maybe not even from Monte Carlo simulations) one can directly load it into Matlab/Octave and apply the post-processing functions. Similarly, an easy way to link the SAFE Toolbox to an external simulation model is to perform the sampling in Matlab, save the input sample X into a text file, run the model against the sampled inputs outside Matlab, load the output samples from the model output file into Matlab, and move on to the post-processing step (see also asterisk in Fig. 2). Advice on how to do this, with a practical example, is given in the user documentation through a specific workflow script.

3. Outlook

SAFE is a modular, flexible, open-source Matlab toolbox for GSA. Its main features are that it facilitates the application of multiple GSA methods, that it includes functions to analyse the convergence

and robustness of sensitivity indices for all methods (including some like EET and RSA where this has not yet become an established practice), and that it provides several visualisation tools for both the investigation of sensitivities and their effective communication. It provides both tools and practical guidelines (through workflow scripts) to assist non-specialist users in performing GSA. At the same time, it is a fully commented code that more experienced users can customise, share and further develop.

The SAFE Toolbox is freely available from the authors for noncommercial research and educational purposes. A website has been set-up to facilitate the Toolbox distribution (bristol.ac.uk/cabot/resources/safe-toolbox/). New releases will be progressively uploaded on this website as new methods for sampling, post-processing and visualisation will be implemented, and registered users will be notified about new releases. By “releasing early, releasing often” (Raymond, 1999), we aim at establishing a tight feedback loop with users of the SAFE Toolbox. Users are welcome to send their feedbacks about the Toolbox, though we do not plan to establish a collaborative software development project at this

```

% Number of uncertain parameters subject to SA:
M = 5 ;
% Parameter ranges (from literature):
xmin = [ 0 0 0 0 0.1 ];
xmax = [400 2 1 0.1 1 ];
% Parameter distributions:
DistrFun = 'unif' ;
DistrPar = cell(M,1);
for i=1:M; DistrPar{i} = [ xmin(i) xmax(i) ] ; end
% Name of parameters (will be used to customize plots):
X_labels = {'Sm','beta','alfa','Rs','Rf'} ;

% Define output:
myfun = 'hymod_nse' ;

% Step 3 (sample inputs space)

r = 100 ; % Number of Elementary Effects
% [notice that the final number of model evaluations will be equal to
% r*(M+1)]

% option 1: use the sampling method originally proposed by Morris (1991):
% L = 6 ; % number of levels in the uniform grid
% design_type = 'trajectory'; % (note used here but required later)
% X = Morris_sampling(r,xmin,xmax,L); % (r*(M+1),M)

% option 2: Latin Hypercube sampling strategy
SampStrategy = 'lhs' ; % Latin Hypercube
design_type = 'radial';
% other options for design type:
% design_type = 'trajectory';
X = OAT_sampling(r,M,DistrFun,DistrPar,SampStrategy,design_type);

% Step 4 (run the model)
Y = model_evaluation(myfun,X) ; % size (r*(M+1),1)

% Step 5 (Computation of the Elementary effects)

% Compute Elementary Effects:
[ mi, sigma ] = EET_indices(r,xmin,xmax,X,Y,design_type);

% Plot results in the plane (mean(E),std(E)):
EET_plot(mi, sigma,X_labels )

% Use bootstrapping to derive confidence bounds:
Nboot=100;
[mi,sigma,EE,mi_sd,sigma_sd,mi_lb,sigma_lb,mi_ub,sigma_ub] = ...
    EET_indices(r,xmin,xmax,X,Y,design_type,Nboot);

% Plot bootstrapping results in the plane (mean(E),std(E)):
EET_plot(mi,sigma,X_labels,mi_lb,mi_ub,sigma_lb,sigma_ub)

% Repeat computations using a decreasing number of samples so as to assess
% if convergence was reached within the available dataset:
rr = [ r/5:r/5:r ] ;

```

Fig. 4. Example of workflow script: (part of) the Matlab script for the application of the Elementary Effects Test to the 5-parameter rainfall-runoff Hymod model.

stage. Hopefully, the SAFE Toolbox and website will contribute towards improving the diffusion and quality of GSA practice in the environmental modelling community.

Acknowledgements

F. Pianosi and T. Wagener are supported by the Natural Environment Research Council [Consortium on Risk in the Environment: Diagnostics, Integration, Benchmarking, Learning and

Elicitation (CREDIBLE); grant number NE/J017450/1]. F. Sarrazin is supported by University of Bristol Alumni Postgraduate Scholarship.

References

- Andres, T., 1997. Sampling methods and sensitivity analysis for large parameter sets. *J. Stat. Comput. Simul.* 57 (1–4), 77–110.
- Brewer, C., 2013. www.colorbrewer2.org. (last accessed 04.10.14.).

- 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.
- Gan, Y., Duan, Q., Gong, W., Tong, C., Sun, Y., Chu, W., Ye, A., Miao, C., Di, Z., 2014. A comprehensive evaluation of various sensitivity analysis methods: a case study with a hydrological model. *Environ. Model. Softw.* 51 (0), 269–285.
- Herman, J., 2014. Sensitivity Analysis Library (Salib) (last accessed 09.02.15.). <http://jdherman.github.io/SALib/>.
- JRC, 2014. Sensitivity Analysis Repository (last accessed 29.01.15.). ipsc.jrc.ec.europa.eu/?id=756.
- Morris, M., 1991. Factorial sampling plans for preliminary computational experiments. *Technometrics* 33, 161–174.
- Pappenberger, F., Beven, K., Ratto, M., Matgen, P., 2008. Multi-method global sensitivity analysis of flood inundation models. *Adv. Water Resour.* 31 (1), 1–14.
- Pianosi, F., Wagener, T., 2015. A simple and efficient method for global sensitivity analysis based on cumulative distribution functions. *Environ. Model. Softw.* 67, 1–11.
- Pujol, G., Iooss, B., Janon, A., et al., 2014. Sensitivity Analysis Package (last accessed 29.01.15.). cran.r-project.org/web/packages/sensitivity/index.html.
- Raymond, E.S., 1999. The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. O'Reilly Media.
- Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S., 2008. Global Sensitivity Analysis, the Primer. Wiley.
- Spear, R., Hornberger, G., 1980. Eutrophication in peel inlet, ii, identification of critical uncertainties via generalized sensitivity. *Water Resour. Res.* 14, 43–49.
- Tang, Y., Reed, P., Wagener, T., van Werkhoven, K., 2007. Comparing sensitivity analysis methods to advance lumped watershed model identification and evaluation. *Hydrol. Earth Syst. Sci.* 11, 793–817.
- Wagener, T., Kollat, J., 2007. Numerical and visual evaluation of hydrological and environmental models using the Monte Carlo analysis toolbox. *Environ. Model. Softw.* 22 (7), 1021–1033.
- Wagener, T., McIntyre, N., Lees, M., Wheater, H., Gupta, H., 2003. Towards reduced uncertainty in conceptual rainfall-runoff modelling: dynamic identifiability analysis. *Hydrol. Process* 17, 455–476.
- Yang, J., 2011. Convergence and uncertainty analyses in monte-carlo based sensitivity analysis. *Environ. Model. Softw.* 26, 444–457.
- Ziehn, T., Tomlin, A., 2009. GUI-HDMR — a software tool for global sensitivity analysis of complex models. *Environ. Model. Softw.* 24 (7), 775–785.