Environmental Modelling and Software 124 (2020) 104588

Contents lists available at ScienceDirect

Environmental Modelling and Software

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

Certain trends in uncertainty and sensitivity analysis: An overview of software tools and techniques

Dominique Douglas-Smith [a], Takuya Iwanaga [a,*], Barry F.W. Croke [a,b], Anthony J. Jakeman [a]

[a] The Integrated Catchment Assessment and Management Centre, Fenner School of Environment and Society, The Australian National University, Canberra, Australia
[b] Mathematical Sciences Institute, The Australian National University, Canberra, Australia

ARTICLE INFO

Keywords:
Uncertainty analysis
Sensitivity analysis
Analysis software packages
Bibliometric trends
Environmental modeling

ABSTRACT

Uncertainty and sensitivity analysis (UA/SA) aid in assessing whether model complexity is warranted and under what conditions. To support these analyses a variety of software tools have been developed to provide UA/SA methods and approaches in a more accessible manner. This paper applies a hybrid bibliometric approach using 11 625 publications sourced from the Web of Science database to identify software packages for UA/SA used within the environmental sciences and to synthesize evidence of general research trends and directions. Use of local sensitivity approaches was determined to be prevalent, although adoption of global sensitivity analysis approaches is increasing. We find that interest in uncertainty management is also increasing, particularly in improving the reliability and effectiveness of UA/SA. Although available software is typically open-source and freely available, uptake of software tools is apparently slow or their use is otherwise under-reported. Longevity is also an issue, with many of the identified software appearing to be unmaintained. Improving the general usability and accessibility of UA/SA tools may help to increase software longevity and the awareness and adoption of purpose-appropriate methods. Usability should be improved so as to lower the "cost of adoption" of incorporating the software in the modelling workflow. An overview of available software is provided to aid modelers in choosing an appropriate software tool for their purposes. Code and representative data used for this analysis can be found at https://github.com/frog7/uasa-trends (10.5281/zenodo.3406946).

1. Introduction

Computational modeling has become a key activity in many areas of research. In the environmental sciences the amount of available computational power and speed has led to the development of environmental models with ever-increasing level of detail and complexity. In this context complexity is reflected by the number of parameters a model incorporates as inputs. These parameters may also be referred to as ‘parameter factors’, ‘factors’ or simply ‘inputs’ in the literature (Norton, 2015). Increasing the number of parameters allows for a more detailed representation of the investigated system while also increasing computational cost and model complexity at an exponential rate. Increased detail (and thus complexity) may reduce the identifiability of parameters - the ability to apportion model results to specific parameter values – but is not always justified or necessary with respect to the aims of the modeling exercise.

Increased complexity has led modelers to better appreciate the issue of model identifiability (Guillaume et al., 2019) and to recognize the importance of understanding the contribution of model inputs with respect to model performance and purpose. Uncertainty and Sensitivity Analysis (UA/SA) refer to the methods and approaches used to help researchers better understand the relative importance of each parameter factor within a given problem context. Put simply, “[S]ensitivity analysis assesses how variations in input parameters, model parameters or boundary conditions affect the model output” (Bennett et al., 2013). With these approaches, it is possible to better understand how *sensitive* model results are to parameter factors and how *uncertain* the model results are (Saltelli et al., 2019; Saltelli and Annoni, 2010). Individual parameter factors may influence one or more outputs and could (conditionally) affect the importance of other factors; referred to as parameter interaction. The practice of analyzing uncertainty and sensitivity is now considered standard modeling practice. The interested reader is directed to (Bennett et al., 2013; Norton, 2015; Pianosi et al., 2016; Razavi and Gupta, 2015) for introductory overviews and further information.

Understanding the relative ‘sensitivity’ of parameters can aid in the

* Corresponding author.
E-mail address: iwanaga.takuya@anu.edu.au (T. Iwanaga).

https://doi.org/10.1016/j.envsoft.2019.104588
Received 13 June 2019; Received in revised form 13 September 2019; Accepted 13 November 2019
Available online 18 November 2019
1364-8152/

Table 1
Descriptions of common UA/SA sampling and analysis techniques and key texts. Qualitative assessment and indications of sampling requirements are provided. Where indicated, p refers to the number of parameters and N the number of parameter sets.

Name	Abbreviation	Description	More information
One-at-a-time SA	OAT	Each parameter is perturbed from its baseline point. May also be known as variations of the name such as one-factor-at-a-time, one-variable-at-a-time, etc. Required number of model evaluations range from $p+1$ to $(N \times p)+1$ where p is the number of parameters and N is the number of desired perturbations.	Czitrom (1999)
Derivative-based SA	–	Family of methods that take partial derivatives of each input parameter with respect to the output.	(Helton, 1993; Norton, 2015)
Variance-based SA	–	Family of methods that attempt to map statistical properties of the output distribution to the inputs used – how variance in the inputs explains variance in the outputs. Variance-based approaches may require an exponentially increasing number of N samples with increasing p to obtain reliable results (e.g. Razavi and Gupta, 2016).	(Norton, 2015; Pianosi et al., 2015)
Monte Carlo sampling	MC	Random sampling: statistically independent samples of the parameter space. Used for variance-based SA.	(Fedra, 1983; Metropolis and Ulam, 1949)
Latin Hypercube sampling	LHS	The range of each parameter in the parameter space is partitioned into N equal-probability divisions. One sample is taken from each of the N partitions, generating N samples per parameter, and a sample from each set of N samples is chosen for each parameter. The process is repeated to obtain the desired number of samples (Norton, 2015).	McKay et al. (1979)
Importance/stratified sampling	–	Estimate the probability density of the parameters, usually uniform or Gaussian, and determine the importance of resulting outcomes in order to define regions in the parameter space. Each region is given an equal quota of randomly distributed samples. Used for variance-based GSA.	Castaings et al. (2012)
Morris method	Morris	An elementary effects method, derivative-based GSA. Ranks parameters by influence on output and non-linearity. Each parameter is stepped along trajectories. The starting points are random and uniformly distributed. The parameters are perturbed once in succession along the trajectory, in random order. The resulting sample consists of the changes in model output caused by each parameter's perturbation (Norton, 2015). Several variations have been proposed (Pianosi et al., 2016) and convergence of sensitivity indices are said to occur with a relatively small number of model evaluations and so is commonly used for factor screening (Gan et al., 2014; Sun et al., 2012). It requires $N(p+1)$ model evaluations, where typically $N \approx p$ or less (Norton, 2015)	(Campolongo et al., 2011; Morris, 1991)
Derivative-based Global Sensitivity Measure	DGSM	Derivative-based GSA. Sensitivity indices are computed by taking the integral over the function domain of the square of the partial derivatives of each factor with respect to the function (Sobol and Kucherenko, 2009). The method is described as being effective at screening parameters for high-dimensional models with low sample sizes (Becker et al., 2018).	Sobol and Kucherenko (2009)
Sobol' method	Sobol'	Variance-based GSA. An MC-based method: analyzes how the variability of a parameter or combination of parameters influences the variability of the output.	Sobol (1993)
Fourier Amplitude Sensitivity Test	FAST	Variance-based GSA. By estimating the output by a sum of sinusoids, each parameter becomes a function of a chosen variable ranging from $-\pi$ to π. Rather than computing the variance and mean of the output as multiple integrals, these now become a single integral with respect to the chosen variable. Variations include eFAST and Random Balance Design (RBD).	FAST: (Cukier et al., 1973) eFAST: (Saltelli et al., 1999; Wang et al., 2013) RBD: (Tarantola et al., 2006)
Distributed Evaluation of Local Sensitivity	DELSA	Derivative- and variance-based GSA. An elementary effects method. DELSA is a multiple starts perturbation method in which squared finite differences are the metric of sensitivity. DELSA is said to be able to obtain the full distribution of sensitivities at a lower cost compared to the Sobol' method (i.e. a comparatively lower N).	Rakovec et al. (2014)
Regression- and correlation-based SA	–	Statistical-based GSA. The sensitivity metric is the regression/correlation coefficient between the input parameters and output after Monte Carlo sampling.	(Iman and Helton, 1988), (Saltelli and Marivoet, 1990)
Regional SA	RSA	Statistical-based GSA. A binary split of the input parameters from a Monte Carlo sample is determined by whether the resulting output respective to an input sample exhibit required behaviors. A cumulative distribution function is applied to the non-behavioral input samples as a metric of sensitivity. Said to have low computational requirements (Sun et al., 2012).	(Spear and Hornberger, 1980; Young et al., 1978)
Generalized Likelihood Uncertainty Estimate	GLUE	Model results are given as probability distributions of possible outcomes. Assesses how accurate these model results are as a representation of uncertainty.	Beven and Binley (1992)
Emulators	–	A simplified model is fit to a sample in order to give a general indication of parameter sensitivity.	(Crestaux et al., 2009; Oakley and O'Hagan, 2004; Oladyshkin and Nowak, 2012; Ratto and Pagano, 2010; Storlie and Helton, 2008)

development of better monitoring strategies and experiment design, for example indicating the priority and amount of data to be collected (Saltelli and Tarantola, 2002). The practice of SA can also help to constrain the parameter space by identifying parameters that may be 'insensitive' or 'inactive', having little to no effect on model results, at least for the purpose of the modeling. Identifying such parameters can help constrain model complexity which in turn eases the computational cost of model evaluations, for example to facilitate uncertainty analysis and the development of surrogate models.

In recent years a wide variety of software tools to support UA/SA processes have become available that make such analyses more accessible to modelers. To gain an overview of the available methods and tools, we applied a hybrid bibliometric approach using publications from the Web of Science database. While reviews of sensitivity analysis practice have been published (see for example Ferretti et al., 2016; Saltelli et al., 2019) and comparisons between UA/SA methods conducted (for example Gan et al., 2014; Sun et al., 2012), to our knowledge there does not appear to be an overview of the available UA/SA tools currently in use across different platforms and programming languages. This paper follows on from and is distinguished from existing reviews (such as Matott et al., 2009; Refsgaard et al., 2007) as it surveys UA/SA in environmental modeling, with a specific focus on SA. We then provide information on the available tools, as revealed through the bibliometric analysis and expert knowledge, including implemented UA/SA methods, programming language, and software features. The aim here is then to provide 1) a brief introduction to the field of UA/SA and its relevance to environmental modeling for those new to the field, 2) an overview of UA/SA research trends, and 3) a guide to the development trends of UA/SA tools, their availability, and relevance.

2. Key UA/SA terminologies and methods

Often the first hurdle for those new to a research area is to grasp the multitude of acronyms and terms used. In this section we briefly outline some common terminology, UA/SA methods, and relevant publications for further reference. These are provided here to contextualize the analysis and discussion later in this paper. The information provided in this section is not exhaustive. Interested readers are directed to Norton (2015) for a more thorough introduction to UA/SA, the descriptions of sensitivity analysis methods in Pianosi et al. (2016), the citations in Table 1, and the citations in (Bennett et al., 2013, p. 3).

Pianosi et al. (2015) identify three stages in a sensitivity analysis: selecting a sample of input values from the variability space, running a model evaluation against these input values, and applying a sensitivity analysis method to the input/output samples to compute *sensitivity indices*, i.e. values which indicate each parameter's sensitivity. For more information about the calculations for various sensitivity indices, see Norton (2015). Here, the *variability space* refers to all possible combinations of values that can be assigned to a model's input parameter set. By running the model with the values sampled from the variability space and taking note of the resultant outputs, analyses can be conducted to calculate the influence that a specific input, or set of inputs, may have, i.e. their sensitivities. The focus of this paper is on providing an overview of tools that aid in conducting these analyses.

Methods to select the sample of input values are often characterized as being either 'local' or 'global'. Global methods (GSA) consider all dimensions of a model "in one grand exercise" (Leamer, 1985), achieved by varying all parameter values at the same time. GSA methods are themselves commonly categorized as being statistical, derivative, or variance based. Statistical methods use statistical analysis of the parameter space as a measure of sensitivity (Pianosi et al., 2016). Derivative-based methods provide indices which characterize the distributional properties of partial derivatives (Razavi et al., 2019). Variance-based approaches determine how different factors contribute to model variance by analyzing and decomposing the variance in model outputs (Razavi et al., 2019). For brevity, a full exploration of these methods is not provided here, but a brief overview, with references to relevant papers, is given in Table 1.

The strength of GSA methods is that they provide a more robust depiction of model uncertainty by comprehensively accounting for parameter interactions (Saltelli and Annoni, 2010). Such approaches assume a random distribution of output values in the parameter space and that such a distribution is plausible. GSA methods can also be computationally expensive to perform as the parameter space being explored can be very large. Sampling methods ('schemes') are used to aid in limiting the number of model runs involved whilst adequately representing the parameter space. The computational cost of applying GSA methods may explain, at least in part, why their use is relatively uncommon compared to their local counterparts.

Local SA methods (LSA) are anchored around a particular point in the parameter space with analysis involving comparisons against a known 'baseline' output (Razavi and Gupta, 2015). The simplest, most naïve, and most common, method of SA is one-at-a-time (OAT). As the name suggests, this approach involves changing the value of a single parameter factor at a time (referred to as 'perturbing') whilst keeping all other parameters constant at their nominal values. This approach could be described as taking samples along a single dimension with the changes to the output then attributed to the factor that was modified. There are different approaches to how much the parameter value is perturbed but often a proportional increment is used – e.g. increase or decrease a parameter by 10% of the nominal value up to and including a given bound (Razavi and Gupta, 2015).

Other LSA methods examine the partial derivatives of output with respect to each input parameter. These are computed at one point in the sample space to determine sensitivity indices. The simplicity of the procedure is advantageous, as well as being computationally inexpensive for first order derivatives as they often do not require a formal sampling approach. Monte Carlo (MC) – a simple random sampling – is commonly used, although it offers a limited representation of the total parameter space (Gan et al., 2014). The downside is that LSA only provides a robust indication of sensitivity for linear or additive models (Saltelli and Annoni, 2010): they do not account for parameter interactions and become computationally expensive when higher order and non-linear effects are considered. To resolve this issue several other sampling approaches have been developed and applied. Given each method and approach have their pros and cons, multiple methods could be applied to obtain complementary results à la ensemble analysis (Sagi and Rokach, 2018) and should be considered where appropriate (Sun et al., 2012). Brief descriptions of commonly employed methods are given in Table 1. Methods are taken to be "common" where they are indicated to be so in recent review papers (specifically, Gan et al., 2014; Pianosi et al., 2016), the references found within these, and those found within the identified corpora (detailed in the next section).

3. Method: The hybrid bibliometric approach

To conduct this bibliometric review a collection of publications (the 'corpora') was gathered from Clarivate Analytics' Web of Science (WoS) database using the available web-based Application Programming Interface (API). Use of the API enabled programmatic access to the publication data and metadata including titles, abstract text, author-supplied keywords, and DOIs. Data was retrieved with the use of Wosis (Web of Science Analysis), a Python package developed to simplify the process of querying the WoS database and aid in data analysis and visualization (Iwanaga and Douglas-Smith, 2019). Publications in the resulting corpora were taken to represent the field of uncertainty and sensitivity analysis in the overarching field of environmental modeling.

To ensure as much transparency as possible, much of the data collection and subsequent analysis was conducted programmatically in the Python programming language. The complete dataset cannot be made available as it is subject to Clarivate Analytics' license terms. Representative datasets are provided instead, along with the code developed for the analysis; these can be viewed as a collection of Jupyter Notebooks and associated files at https://github.com/frog7/uasa-trends (Douglas-Smith and Iwanaga, 2019). Names of specific notebooks will be referred to throughout this text where further detail can be found.

The corpora was iteratively and incrementally refined through a semi-autonomous process of topic identification, keyword search, and subsequent manual analysis of the publications with the aid of key phrase extraction. Topic modeling (briefly described in Section 3.2) was used to aid in identifying a collection of papers relevant to uncertainty and sensitivity analysis and their overarching focus, be it an application of or guiding frameworks for UA/SA. The publication and citation trends within these topic areas were then analyzed. Additional topic modeling, complemented by a keyword search process, was used to identify papers related to the use of UA/SA software. These were manually combed through with the aid of an automated key phrase identifier that helped to reduce the amount of text to be examined. A subset of these papers were investigated for mention of software tools and packages. The general search and analysis approach is depicted in Fig. 1, with further detail on topic modeling and key phrase identification provided within this section.

3.1. Initial search

The initial corpora for the analysis was identified by specifying the search phrase (with search fields bolded): **TS**=("sensitivity analysis" OR "uncertainty analysis" OR "uncertainty quantification" OR "uncertainty propagation" OR "local sensitivity analysis" OR "LSA" OR "one-at-a-time" OR "exploratory modeling" OR "OAT" OR "global sensitivity analysis" OR "GSA" OR "all-at-a-time" OR "AAT") AND **WC**=("ENVIRONMENTAL SCIENCES" OR "WATER RESOURCES" OR "ENGINEERING ENVIRONMENTAL" OR "INTERDISCIPLINARY APPLICATIONS"). This returns publications that use at least one of the specified terms (those listed for the "**TS**" field) within the title, abstract, or author supplied keywords for publications in the the WoS defined subject areas; specified for the "**WC**" field. The raw search string is supplied for transparency and can be used to obtain the corpora from WoS.

Only English language publications between 2000 and 2017 were considered for this study, with the ending year selected as the data request occurred in December of 2018. The approach taken at the time was to include full year datasets only. The final search phrase applied with the specified time frame reduced the number of matches from over 500 000 to 11 718 publications. The number of results obtained through the unrestricted search were far too many to comprehensively review, at least in a timely manner. The initial corpora for this study (of 11 718 publications) were then further constrained through the process depicted in Fig. 1 and is described in more detail below.

3.2. Topic identification

A key focus in this study is the software tools and packages available to support UA/SA processes, the methods they implement, and the trends of these. To this end, topic modeling was applied to constrain the corpora to relevant publications for further consideration. Topic models attempt to cluster texts into similar or related topics based on commonly occurring words and can aid in identifying new and emerging fields whilst also reducing the likelihood of bias and the required hours for a systematic review (Achakulvisut et al., 2016; Westgate et al., 2018). Topic modeling has been applied before to reduce the time and difficulties encountered when conducting systematic reviews (Westgate and Lindenmayer, 2017), however their use is still relatively limited and perhaps underutilized. Although software is available to aid in these bibliometric approaches, currently no single software package provides all necessary functionality to conduct end-to-end systematic mapping – the classification of articles based on their contents – of research literature from data collection through to summarization and visualization. Arguably the conjunctive application of systematic mapping and bibliometric analysis is still in its infancy (as evidenced by Nakagawa et al., 2018).

Topics are identified by the common co-occurrence of semantics within a discipline. For example, "sensitivity" in the context of SA would conceptually be expected to appear in texts containing words such as "analysis", "uncertainty", and "modeling". The term "sensitivity" may also appear in relation to physical/psychological response to stimuli, in which case the term will appear alongside terms associated with the medical and therapy fields. Topics can be identified and represented through their common semantics. The topic modeling approach provided within Wosis – Non-negative Matrix Factorization (NMF) – is implemented with the scikit-learn Python package (Pedregosa et al., 2011). The approach allows publications to be assigned to one or more topics (Arora et al., 2012) and has been shown to be appropriate for collections of short texts (Chen et al., 2019). This process was complemented with a traditional keyword search to help identify publications related to specific subjects.

Tokens, meaning specific words or terms, for topic modeling consisted of the text found in the document titles, abstracts, and keywords. The top 1000 tokens found within the corpora based on Term Frequency-Inverse Document Frequency (TF-IDF) rankings were selected for topic modeling. TF-IDF is a common ranking method used in text mining (Beel et al., 2016). A high TF-IDF score indicates that the word token has a high frequency within specific document(s), but a low number of occurrences within the entire corpora. Weighting the score in such a manner has the effect of filtering out commonly used tokens which may not have high semantic importance.

3.3. Key phrase identification

Once a topic area is identified, the resulting sub-corpora can be further constrained through automated key phrase identification. The approach summarizes text, aiding reviewers to identify irrelevant publications by reducing the amount of text for manual review. The implemented approach attempts to identify these phrases of interest by scoring sentences based on their similarity with other sentences throughout the abstract text.

To elaborate, each sentence (s_i) is compared with other sentences in the abstract (s_y), which are initially filtered based on the presence of a root token which is taken to be the token that appears in the middle of s_i. This root token selection approach is used in Rabby et al. (2018) for its simplicity and computational efficiency. The similarity between s_i and s_y is then scored based on the ratio of the intersection of the two sentences. Sentences with three or less tokens (i.e. words, numbers, or other counted by splitting the text on individual spaces) are ignored.

The approach assumes that important features of the publication, such as its key findings, will be repeated throughout the considered fields (title, abstract, and author-supplied keywords). These may, for example, be introduced or alluded to, framed and the implications discussed. The implemented approach is therefore dependent on the abstract length, with longer texts preferred. Poor performance can be expected for very short abstracts (e.g. 3 sentences or less) and these were ignored for the purpose of this study. Comparisons with an established key phrase identification approach, RAKE: Rapid Automatic Keyword Extraction (Rose et al., 2010), implemented through the rake-nltk Python package – indicate that the above approach produces, subjectively, key phrases that were more useful for the purpose of this study (see Table 2).

3.4. Citation and trend analysis

Citation analysis indicates the papers being referred to by other papers within the corpora as well as the overall number of citations the given publication has received, the assumption here being that impactful papers are more likely to be cited. The number of citations is then used to indicate papers that are of high importance to the subject at hand. Both the total number of citations and the average citations since publication were used in the analysis. Publication trends within topic areas aided in identifying the general focus and direction taken by the research community. Plotted publication trends were used for this

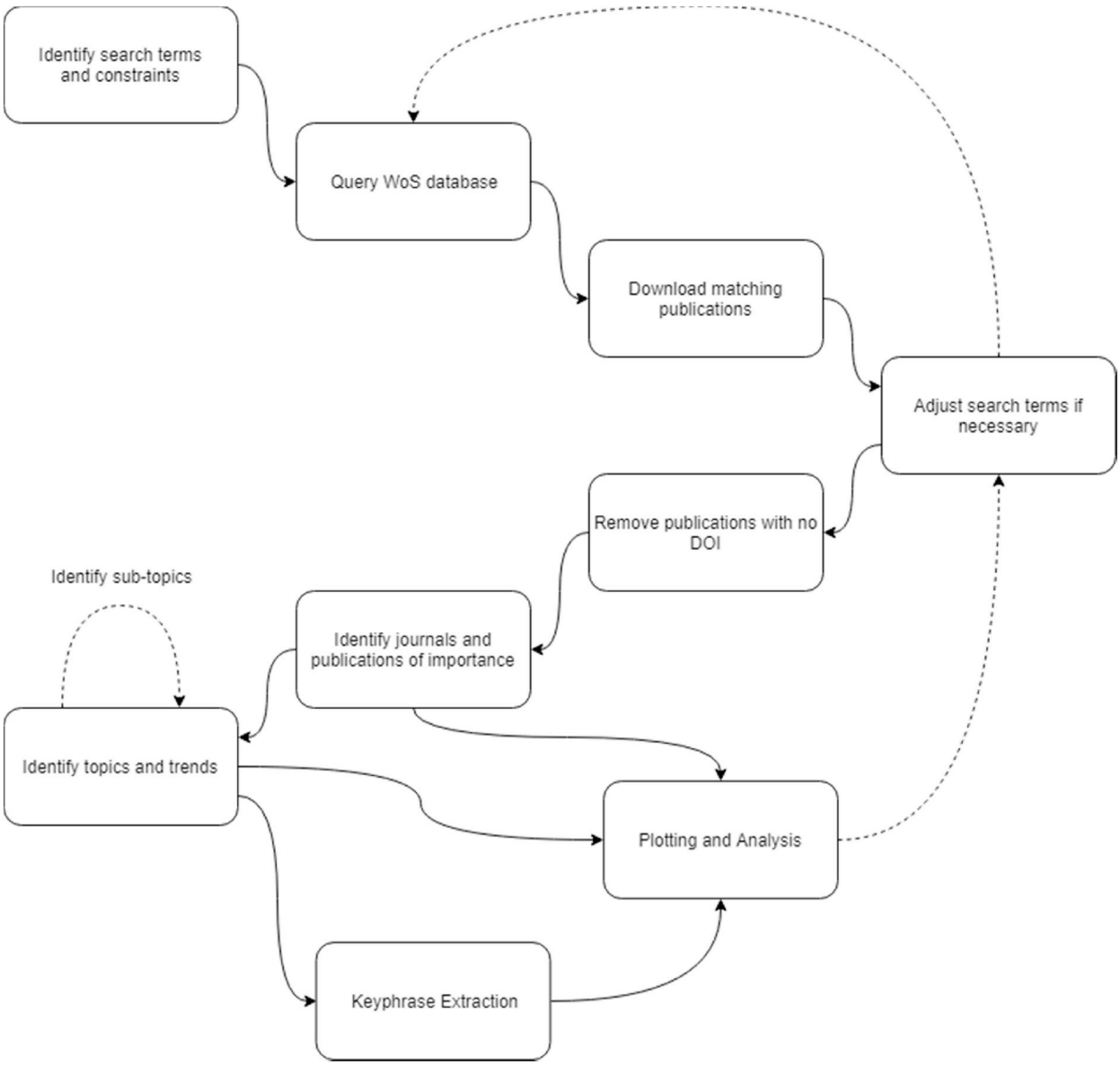


Fig. 1. The hybrid bibliometric analysis process. Identification and subsequent analysis of the final corpora followed an iterative process through which publications were progressively filtered to arrive at a relevant subset.

purpose.

4. Results: UA/SA packages

Of particular interest to this paper were the trends of software packages implementing UA/SA methods and these are discussed here. The final corpora was broadly categorized into two topics – "Applications" and "Frameworks" using the topic model described in the "Method" section. Publications focused on UA/SA frameworks and guidelines were placed into the "Frameworks" sub-corpora, while "Applications" included those taken to be focused on the application of UA/SA methods. From each of these a keyword search was applied to identify publications related to model sensitivity, optimization, uncertainty quantification, or toolboxes, in order to build a sub-corpora related to the software.

Manually sorting the identified publications with the aid of the automated key phrase extraction tool reduced the corpora to 193 papers (referred to as the "Software corpora", see Notebook 5c "Software packages analysis"). Papers were regarded as relevant if they: included direct reference to UA/SA or optimization software packages; were theory, review, or framework papers that recommended software implementation to a given field; or referred to other methods and packages of interest to expert opinion. Further detail and a general bibliometric overview are provided in a later subsection.

There does not appear to be a strong correlation between the Applications and Software corpora (Fig. 2). The Software corpora has a stable publication trend relative to those focusing on applications over the surveyed timeframe. A spike in publications in 2007 proportional to the full final corpora can be seen (Fig. 3). While publications on the software for UA/SA have been increasing (see Fig. 4) the trend relative to the Applications corpora and the full corpora could be indicative of 1) a general ambivalence towards reporting use, or development of, general UA/SA software, 2) a common set of UA/SA software, 3) a reliance on self-coded analysis software, or 4) increased tendency to release software in a directly citable manner, e.g. with an attached DOI which the WoS database does not include but this is considered unlikely however in the authors' opinion.

The slow uptake of software packages, relative to the Applications corpora, could also be due to 1) a lack of documentation for beginner users and 2) a lack of awareness of available software packages. In the first case, beginner users may not use software that requires significant learning time for effective use, especially when no clear user guide, examples to draw from, or community to engage with exists. In the latter case, modelers should be made aware of the available software that can reduce the time required to conduct UA/SA and promote better practices in UA/SA.

The software evident in the literature range from those specific to a field, general-purpose packages, to custom-made code. Fields such as hydrology, climate, chemistry, and more general environmental modeling and engineering used field-specific packages. A complete list of reviewed software publications and their related software packages can be found in Notebook 5a “Finding software packages by keyphrase extraction”. The most common analysis method provided by UA/SA software was found to be Sobol’ with the R sensitivity package providing the widest mix of methods (see Table 3). Surveyed software tools typically did not provide OAT analysis, perhaps due to its simplicity or a sign of its decline. Publication of software related papers is relatively stable, with a proportional spike in 2007 (Fig. 3).

Software for the development of emulators did not feature heavily within the Software corpora although they are present, the HDMR method being one example (described later on). Software to develop emulators include ChaosPy (Feinberg and Langtangen, 2015), the PRISM Uncertainty Quantification framework (Hunt et al., 2015), GTApprox (Belyaev et al., 2016) and UQ-PyL (Wang et al., 2016). A collection of functions presented as a Matlab toolbox is also introduced in Vu-Bac et al. (2016). All of these with the exception of Vu-Bac et al. (2016) were developed in the Python programming language. Application of Artificial Neural Networks and similar approaches did appear in the corpora but is not a topic of focus here.

As aforementioned, current trends have shown an increased interest in best practices. Three SA packages, released within the past five years, reflect these changing attitudes: PSUADE (Gan et al., 2014), SAFE (Pianosi et al., 2015), and VARS-TOOL (Razavi et al., 2019). PSUADE (a Problem Solving Environment for Uncertainty Analysis and Design Exploration) provides users with implementations of UQ methods, including sampling techniques and SA methods (both local and global). The package has had general application to various modeling scenarios. SAFE (Sensitivity Analysis For Everybody) provides users with implementations of global SA methods, with the ability to perform multiple SAs, robustness assessment, and convergence analysis without further model runs. As reflected in its name, this package was designed to allow global SA to be accessible to a more general audience.

The most recently released package in the survey, VARS-TOOL, provides implementations of sampling techniques and global SA methods, including derivative-, variance-, and variogram-based, which can all be performed from a single sample. The variogram approach to SA reportedly links both local and global approaches.

4.1. Survey of packages in common programming languages

Brief descriptions of software found in the corpora are provided here, categorized by their implementation language. Some packages may be listed more than once as various implementations may exist, or interoperability between languages is supported. We decided to categorize the packages based on the implementation languages as most packages are not standalone tools with user interfaces ready to be used and are often provided as a library to be incorporated programmatically. Indicating the implementation language also allows readers to identify packages in a familiar language for potential adoption. Very few packages were found to provide a Graphical User Interface (GUI) so some amount of programming ability and experience is the baseline expectation. In the vast majority of cases, users are expected to have a passing familiarity with the UA/SA methods being applied as very little protection against improper use is provided (a further brief discussion is in the Recent developments section). Table 3 and 4 provide summary overviews of the software and packages.

4.1.1. Fortran

Fortran was one of the earliest programming languages available and arguably still dominates the scientific programming landscape. Fortran modules from the surveyed literature are JUPITER API and UCODE. There is also a Fortran repository of UA/SA functions supported by the Joint Research Centre (Pianosi et al., 2015). The JUPITER API (Joint Universal Parameter IdenTification and Evaluation of Reliability Application Programming Interface) attempts to provide a standard set of programmatic functions for developing UA/SA software and serves as the underlying “engine” for other UA/SA packages (UCODE 2005/2014 were developed on top of this API). The provided modules are developed in Fortran-90 and support parallelization and local (derivative) sensitivity analysis. JUPITER API was first released in 2006 and its latest release was 2013. Its affiliated webpage was last updated in 2016, suggesting an active community. It is provided freely and under an open-source license with a user manual and examples of applications.

First released in 1998, UCODE (Universal inverse CODE) was developed in Fortran90, Fortran95, and Perl. It originally implemented inverse modeling methods, and by its first revision (2005) consisted of post-processing modules for (and not limited to) SA, calibration, and UA. The second revision (2014) included MCMC in the UA module and made the platform more compatible with models developed in Matlab or using a GUI. This can be viewed as a response to changing trends in model development, particularly the proliferation of Matlab-based models. User documentation is available for download. Although the software is still available for download, its development has ceased.

Table 2
Example of key phrases identified and extracted by Wosis compared with the RAKE method provided in the ’rake-nltk’ package. The original abstract was taken from Roos et al. (2015). Both RAKE and Wosis approaches were limited to a minimum of 3 words per phrase. Identified phrases are ordered by score and do not follow the original paragraph structure.

Wosis	RAKE
We propose a novel formal approach to prior sensitivity analysis, which is fast and accurate.	hoc modified base prior parameter values parameters within deeper layers
Other formal approaches to prior sensitivity analysis suffer from a lack of popularity in practice, mainly due to their high computational cost and absence of software implementation.	identifiability issues may imply prior sensitivity examination plays detect high prior sensitivities
Despite its importance, informal approaches to prior sensitivity analysis are currently used.	prior sensitivity analysis suffer parametrized Bayesian hierarchical models
This is especially true for Bayesian hierarchical models, where interpretability of the parameters within deeper layers in the hierarchy becomes challenging.	prior sensitivity analysis bayesian hierarchical models quantifies sensitivity without high computational cost
They require repetitive re-fits of the model with ad-hoc modified base prior parameter values.	applied bayesian analyses novel formal approach hierarchy becomes challenging

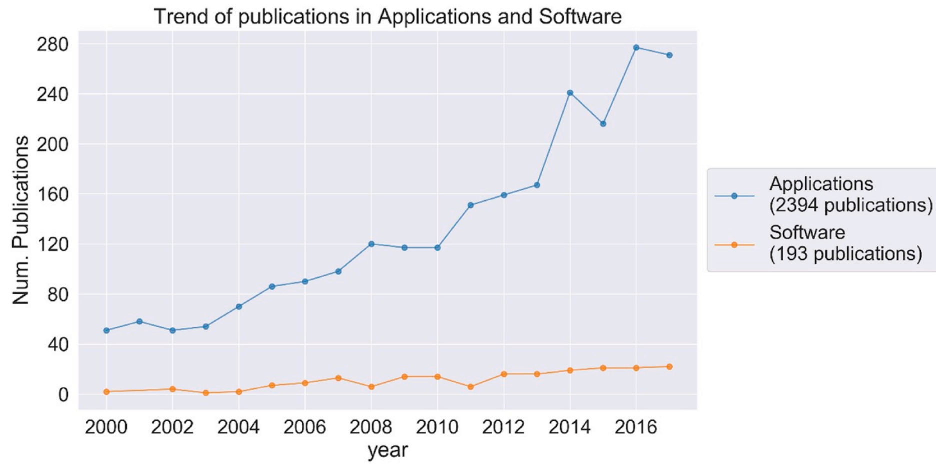


Fig. 2. A comparison of the publication trend of Applications and Software. Both increase during the timeframe, although it is unclear whether the trends are related.

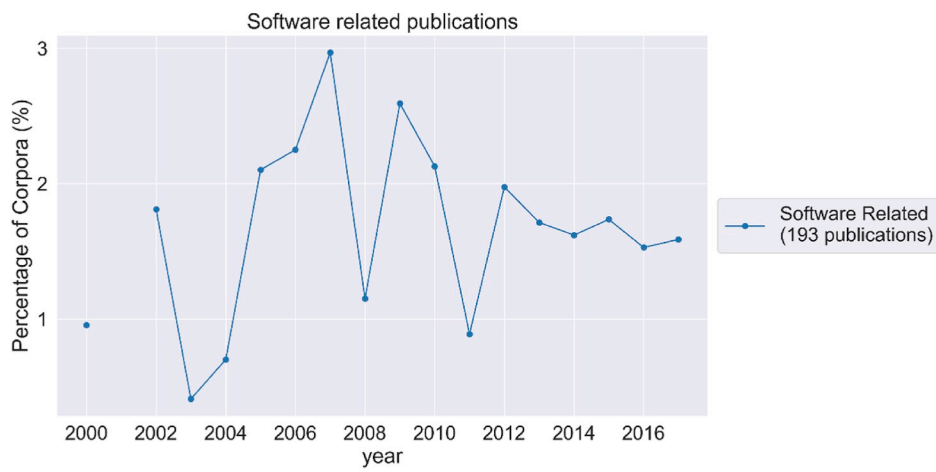


Fig. 3. Relative trend of software publications. The trend is relatively flat but note the initial spike in publications in 2007.

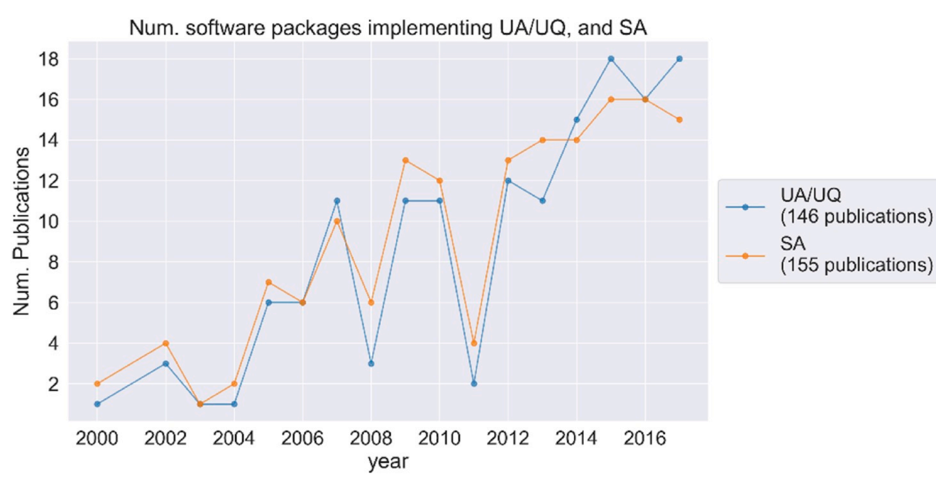


Fig. 4. Absolute publication trends for software packages implementing UA/UQ, and SA methods.

4.1.2. C/C++

Surveyed software available in C/C++ include Dakota, PSUADE, PEST, and VARS-TOOL.

The Dakota toolkit had its initial release in 1994 to provide optimization tools for engineers. With further development, it now includes sampling methods, global SA methods, parameter estimation, and UQ. The software can be tightly-, semi-, or loosely-coupled to the target model, requiring the user in the first two cases to modify their code or use a direct interface. The package is presented as being accessible to beginners and involves advanced features for more competent users. It operates on Linux, Windows, and Unix. Parallelization is possible and there is a GUI option. It is freely available for academic use and open

Table 3
Comparison table of available UA/SA methods in the surveyed packages.

Name[a] (Language)	MC	LHC	Morris	DGSM	Sobol'	FAST	RSA	Regression/ Correlation SA	Other
SimLab (Matlab)		✓	✓		✓	✓		✓	
MCAT (Matlab)					✓			✓	GLUE and many others (UA, parameter estimation)
GUI-HDMR (Matlab)					✓			✓	HDMR Emulation
UQ Lab (Matlab with R plugin)			✓		✓			✓	Bayesian Inversion, Kriging, Support Vector Machines, and more
SAFE (Matlab/R)			✓		✓	✓	✓		Dynamic Identifiability Analysis, PAWN
VARS-TOOL (Matlab, C++, Python, and built into OSTRICH)		✓	✓		✓		✓		STAR-VARS, Generalized Global Sensitivity Matrix
Dakota (C/C++, Fortran77/Fortran90)	✓	✓	✓		✓				Supports emulation and many other UA/SA methods
PSUADE (C++)	✓	✓	✓			✓			Fractional Factorial, Central Composite, Probabilistic methods, and others
PEST/PEST++ (C/C++, Fortran)	✓		✓		✓				
R Sensitivity (R)	✓	✓	✓	✓	✓	✓			DELSA, Kriging, many other variations available
SALib (Python)		✓	✓	✓	✓	✓			Delta Moment Independent Measure, Fractional Factorial, Finite Difference
MADS (Julia, C/C++)	✓	✓			✓	✓			Kriging, Bayesian Information Gap Decision Theory, Support Vector Regression
GANetXL (Excel)									Single- and multi-objective genetic algorithm
UCODE (Fortran90/95, Perl)								✓	
MOUSE (Java)			✓			✓	✓		GLUE
GLUE (R, Matlab)									GLUE
OSTRICH (standalone)								✓	GLUE, user defined evaluations also possible

[a] As documentation can lag behind releases, software may include implementations of methods not listed in the table (at time of writing).

source. A user community exists, including mailing lists and interaction with developers. Documentation includes user manuals, examples, and release notes. Dakota is well maintained, its most recent release and webpage update being in 2018. It is an example of software that has kept up to date with the latest trends in UA/SA and software implementation.

PSUADE (Problem Solving environment for Uncertainty Analysis and Design Exploration) can link to simulation code in any language. It provides users with 14 sampling methods and 12 SA methods, both local and global SA. It was developed for large complex systems models and has been applied to various fields. The software has a free public license and is open source. A collaborative user community exists. The software and documentation (a user manual) are available for web download. The package is well maintained, with its latest release and update in 2018.

PEST (Parameter EStimation Toolkit) is designed primarily for model calibration. Originally released in 2003, and with its most recent release in 2019 it has remained up to date with the latest research in environmental modeling. The current package provides parameter estimation and uncertainty analysis, including Monte Carlo analysis, and has parallelization capabilities. The software is designed for complex environmental models, and other models. Models written in C, C++, Fortran, and Python have interoperable interfaces available. It is free, although the license does not appear to be specified, and well-documented for ease of use. Developer-user interaction is encouraged, and training courses are offered.

4.1.3. Matlab

Identified packages of interest written in Matlab are Simlab, MCAT, GUI-HDMR, UQLab, SAFE, and VARS-TOOL.

SimLab is a package for Monte Carlo-based SA, written in Matlab and supplied by the Joint Research Centre. Initially released in 1985, its latest release was 2008 and its associated webpage was last updated in 2016. It provides Monte Carlo and other random sampling methods, test functions for educational purposes, and GSA (correlation-, regression-, and variance-based). The SA follows a loosely-coupled approach requiring only the model output to be fed in. It is freely available for academic use and open source. No user community appears to exist. The documentation consists of a reference manual and the software is available for web download.

MCAT (Monte Carlo Analysis Toolbox) implements Monte Carlo SA. Its first release was 2001 and a companion paper, highlighting the importance of best practices in SA, was released in 2007. However, no further research appears to have been conducted since this time and links to software download provided in the companion paper have expired. This package is of interest as an example of software tooling designed to promote modeling SA best practices. The software provides implementations of UA/SA methods, including regional SA, Monte Carlo analysis, and GLUE. A GUI was developed for it in 2007. The package is free and open source, and documentation includes a manual and examples. No user community appears to exist, however, there is an unofficial GitHub page (see Table 4).

GUI-HDMR (Graphical User Interface-High Dimensional Model Representation) provides HDMR, a variance-based SA method, which the developers advertise as an alternative to other contemporary SA methods. The user must supply an appropriate sample of the model output (there is a complementary package, RS-HDMR [Random Sampling-HDMR] for this purpose). Users have the choice of using a GUI or a script-based interface. The software is reportedly user-friendly and has been applied to various fields. It is freely available for academic use, but not open source. The software and user documentation are available for web download. Although the related publication is highly cited, this software appears to be abandoned, having its first and last release in 2008. A lack of user community and implementation of a single SA method could be a cause for this.

UQLab (Uncertainty Quantification Laboratory) provides, among

Table 4
Summary details (specifications, usability) of some available software of interest. Dash (–) indicates the information could not be found.

Name and Language	First and Last Release (latest update[a])	License	Community	Docs	Indicated required expertise	Related publication	Link to source/ software	Comments
SimLab (Matlab)	1985, 2008 (2016)	Freely available for academic use, End User free license.	–	Manual and examples	Professional tool for model developers, scientists, and professionals	JRC (2015)	https://ec.europa.eu/jrc/en/samo/simlab	Development and simulation tool for UA/SA No GUI.
MCAT (Matlab)	2001 (2007)	Free and open source	–	Manual and examples		Wagener and Kollat (2007)	Unofficial GitHub page https://github.com/ICHydro/MCAT	Monte Carlo Analysis Toolbox GUI is available
GUI-HDMR (Matlab)	2008	Freely available for academic use, but not open source.	–	Manual		Ziehn and Tomlin (2009)	http://www.gui-hdmr.de/	Software does not appear to be actively developed but still available and in use GUI is available
UQ Lab (Matlab with R plugin)	2014, 2018 (2018)	Free for academic use. Content management system is licensed, scientific modules are open source.	User collaboration encouraged, users can contribute to code with revision by developers	Manuals, examples, release notes.	Beginner to advanced functionality	Marelli and Sudret (2014)	https://www.uqlab.com/download	Tagline states "make uncertainty quantification available for anybody, in the field of applied science and engineering" Plugin for R Sensitivity package available Supports parallelized analysis. Has a GUI
SAFE (Matlab/ R)	2015, 2015 (2018)	Freely available for academic use, open source.	No user community, easily adapted to personal use	Pianosi et al., workflow scripts	Beginner to advanced functionality	Pianosi et al. (2015)	https://www.safetoolbox.info/register-for-download/	Designed for users with limited global SA/ Matlab experience Has various GUIs available
VARS-TOOL (Matlab, C++, Python, and built into OSTRICH)	2016, 2018 (2018)	Free for non-commercial use.	–	Manual	Beginner to advanced functionality	Razavi et al. (2019)	http://vars-tool.com	Supports parallelization
Dakota (C/ C++, Fortran77/ Fortran90)	1994, 2018 (2018)	Freely available for academic use, open source with various levels of user interaction. GNU LGPL from version 5.0.	User mailing list and user-developer interaction.	Manual and examples	For users experienced with UA/SA	Adams et al. (2010)	https://dakota.sandia.gov/content/getting-dakota-source-code	Toolkit for optimization, experimental design, and UA/SA Supports parallelization Has a GUI Linkages with Python, Matlab, and Scilab available
PSUADE (C++)	2013, 2018 (2018)	Free public license, open source, LGPL.	User community	Manual	Said to be beginner friendly	Gan et al. (2014)	https://github.com/LLNL/psuade https://computation.llnl.gov/projects/psuade/software	Supports parallelization
PEST (C/C++, Fortran)	2003, 2019 (2019)	Free	User-developer interaction, training courses	Manual, tutorial		Doherty (2018)	http://www.pesthomepage.org/Downloads.php	Supports parallelization Linkages with Python available
R – Sensitivity package	2006, 2018 (2018)	Free public licence (GPL-2), open source.	Developer community.	Manual	Assumes knowledge of R	Iooss et al. (2018)	https://CRAN.R-project.org/package=sensitivity	
SALib (Python)			User community	Manual, examples,		Herman and Usher (2017)		

(continued on next page)

Table 4 (*continued*)

Name and Language	First and Last Release (latest update[a])	License	Community	Docs	Indicated required expertise	Related publication	Link to source/ software	Comments
	2013, 2018 (2018)	Free public licence (MIT), open source.		release notes	Assumed knowledge of Python		https://github.com/SALib/SALib	Has some visualization methods chiefly for the Morris method
MADS (Julia, C/C++)	2016, 2018 (2018)	Free public licence (GPL), open source.	User community	Manual and examples	Beginner to advanced functionality	Various publications listed under https://mads.lanl.gov/#research	https://github.com/madsjulia/Mads.jl	Supports parallelization
JUPITER API (Fortran90)	2006, 2013 (2016)	Free and open source	User-developer interaction	Manual, examples		http://water.usgs.gov/software/JupiterApi		Application Programming Interface to improve model analysis software development Supports parallelization Used to develop other tools, including UCODE (below)
UCODE (Fortran90/ 95, Perl)	1998, 2015 (2016)	Free and open source	–	Manual		Poeter and Hill (1999)	https://igwmc.mines.edu/ucode/	Software appears to be abandoned but download still available
MOUSE (Java)	2014, 2016	Free and open source	–			Ascough II et al. (2015)		Affiliated webpage unavailable Software not under active development but is being maintained Reportedly has a GUI
GLUE (R, Matlab)	1992, 2013 (2016)	Free for academic use, open source	–	Manual and examples		Beven and Binley (1992)	http://www.uncertain-future.org.uk/?page_id=131	Method implemented in software developed by creators (R) and users (Matlab) with implementations found in other packages
SWAT (Fortran)	2000, 2018 (2019)	Free public licence, open source	User community, user-developer interaction, workshops/ conferences	Manual	May be difficult for beginners	https://swat.tamu.edu/software/	https://swat.tamu.edu/software/plus/	Externally developed tools/interfaces developed to implement e.g. SA, GUI
OSTRICH (standalone)	2017	Free and open source	User community for hydrologists	Manual, examples		Matott (2017)	http://www.eng.buffalo.edu/~lsmatott/Ostrich/OstrichMain.html	Supports parallelization

[a] Latest update refers to last identified date in which documentation or code was released. Documentation refers to user/technical manuals, publications specifically on the software or other.

other tools for UQ, tools for statistical analysis, such as sampling and global SA. Global SA methods are supplied through a linkage with the R Sensitivity Package. Parallelization is supported. The package is user friendly and adaptable to various levels of computational experience. Collaboration amongst users is encouraged and users can contribute to code, with revision by the major developers. It is portable between operating systems and freely available for academic use, however documentation is not freely available. The software is well-maintained, with its latest release and update in 2018.

SAFE (Sensitivity Analysis For Everyone) is compatible with the GNU Octave environment and a version implemented in R exists, making it the most openly accessible of all the surveyed Matlab packages. It runs on any operating system. The toolbox was designed to make global SA accessible to users with limited knowledge of global SA or Matlab, whilst also allowing more advanced users to explore, research, and better understand SA. Users are provided with various sampling methods, local and global SA methods, and a GUI (see Table 3). Although there appears to be no collaborative user community, user-developer interaction is possible via email. The software is freely available for academic use and is open source. Documentation includes the companion paper (Pianosi et al., 2015) and additional information provided in workflow scripts. There have been no recent releases, however the website is maintained (last update 2018).

VARS-TOOL is also available in C++ and OSTRICH (a user-independent interface). It features off-line and on-line mode options for running models in any language or operating system. Numerous sampling and SA methods are supplied, including VARS. It is said to be user-friendly and accessible to various levels. It appears to operate as a command-line interface, without a GUI. Although recently developed, there is no collaborative community. The software is freely available for non-commercial use and is open-source. There are capacities for parallelization and reporting and visualization tools; its documentation consists of a manual.

4.1.4. R statistical language

The main SA package for the R language is the R 'sensitivity' package. Like Python, the R language is widely used in the sciences and so many of the tooling support interoperability with R (see the section on Python below, and Table 4). The R 'sensitivity' package supplies various SA and sampling methods. It offers loose coupling with models implemented in other languages as well as in R. Test cases are supplied for research and comparison purposes. The package requires knowledge of R, which itself is portable between operating systems and freely available. A developer community exists and the available documentation consists of a reference manual. Since its initial release in 2006 more recently developed methods have been implemented and included in its latest release (in 2018).

4.1.5. Python

As with R, users of Python have a large assortment of options generally due to Python being a general-purpose language often used for interoperability across languages (see Table 4). The principal SA package developed in Python appears to be SALib (Sensitivity Analysis Library) which provides global sampling and analysis methods and is distributed under a free public license. Model runs can be invoked directly or separately ("offline"). SALib is most applicable to systems modeling and knowledge of Python is assumed. It is a freely available, open-source package, with a collaborative user community. SALib is well documented and well maintained: documentation includes an installation guide, basic usage guide, a complete module reference, and release notes; its latest release was 2018. SALib supports visualization of Morris results only, although this feature appears to be under-documented. A separate visualization tool is available for analysis of Sobol results called "savvy" (Hough et al., 2016), however this package was not examined in-depth.

4.1.6. Java

There appears to be limited SA packages implemented in Java, at least in the reviewed corpora. A response to this limitation is the MOUSE (Model Optimization, Uncertainty and SEnsitivity Analysis) package. This is an implementation of MCAT and OPTAS model calibration software for modelers using Java. It is indicative of the continued influence of the packages MCAT and OPTAS. Its first release was in 2014 and was last updated in 2016. Although claiming to be free and open-source, we could not find relevant information to access the package.

4.1.7. Julia

MADS (Model Analysis & Decision Support) is an SA package available for the Julia programming language. The analyses it supports can be tightly- or loosely-coupled with an existing model. In the module documentation, extensive information is provided for all functions included in the main module ("Mads.jl"). The documentation details modules and examples and, although extensive, was found not to be user-friendly, with functions and methods often lacking meaningful descriptions. MADS is said to support use in High-Performance Computing (HPC) environments. It is a freely available open-source package with a collaborative user community.

An inherent advantage of MADS is the relative youth of the Julia language, with v1.0 released in 2018. Due to its relative youth, it leverages lessons learnt in older programming languages and was developed with modern computational architecture in mind. This means that concurrent and parallel programs are relatively easy to develop in Julia (Bezanson et al., 2017) and it has had demonstrable success on HPC platforms (see for example Regier et al., 2019). The disadvantage of this youth, however, is that the user community – while growing quickly – is still relatively small compared to that of established languages. As such, the language ecosystem is undergoing continual development and may still be immature.

4.2. Active use and development

To gauge the level of support and active development occurring for each software tool, we attempted to identify websites, evidence of userbases, public code repositories, journal publications which specifically mention the software tool, and other indications of activity. Through this process we found that many of the packages present in the literature are no longer under active development, although the code and software may still be available for use.

A key issue in developing software for UA/SA is longevity. We find that those packages that are currently used and under active development and maintenance have the advantages of being open source, well documented for transparency and ease of use, have an active user-community, and offer implementations of a range of UA/SA methods for general-purpose application as opposed to providing a specific method for a specific model. Packages that have fallen into disuse may still be useful with the caveat that there is no supportive community to rely on (for bug-fixes, troubleshooting, user-support, and so on). Table 3 provides an overview of the available UA/SA methods in the surveyed packages, while details of the software can be found in Table 4.

4.4. Bibliometric overview

The initial corpora from WoS consisted of 11 718 publications from which journals deemed to be unrelated to the topic areas of interest (as specified by the search terms used), journals with less than three identified publications, and those without a valid DOI were removed. The final corpora consisted of 11 625 publications. Knowing that researchers build on prior work and given the exponential growth of published material (Bornmann and Mutz, 2015; Haddaway and Westgate, 2018), we assume in this analysis that the identified corpora is representative of the UA/SA field. Full details of this process can be found in Notebook 2 "Create filtered corpora". The number of publications in the environmental UA/SA field have been increasing at an exponential rate (depicted in Fig. 5) with Journal of Hydrology having the most publications overall and experiencing the largest year-on-year gain within the analyzed time frame (Fig. 6).

To facilitate analysis, the final corpora was broadly categorized into two topic sub-corpora – "Applications" and "Frameworks" – using the topic model. As a reminder, the final corpora represents a collection of UA/SA research. Publications focused on UA/SA frameworks and guidelines were placed into the "Frameworks" sub-corpora, while "Applications" included those taken to be focused on the application of UA/SA methods. The topic model was iteratively applied and key phrases from top-cited papers were qualitatively examined to determine the focus of the publications. The specifics of the undertaken process can be seen in Notebook 4 "UASA topic modeling".

A keyword search was applied within these topic corpora to sort publications further into those relevant to uncertainty quantification (UQ), UA, and SA. The resulting collections contained 1 940, 2 751, and 1 360 publications, respectively. To distinguish between LSA and GSA methods, specific keywords were searched for in the combined corpora, including, for example, "local sensitivity", "OAT", "one-at-a-time" for local methods and "global sensitivity" and "GSA" to indicate global methods. In addition to these, newer SA methods identified through manual inspection of the corpora were also searched for, such as "active subspaces" and "variograms".

4.4.1. Trends and directions

As suggested by the general publication trends (in Fig. 7), all topics (UA, SA, Frameworks, and Applications) saw large increases in the absolute number of publications over the 2000–2017 timeframe. Within the same time period the proportional share of the filtered corpora has declined for SA (by 4.5%), while UA has increased (by 5%), which may indicate a gradual shift towards being more inclusive of uncertainty related matters in analyses as well as a general need for uncertainty guidelines in environmental modeling (see Notebook 4 "UASA topic modelling").

The five most active journals in the Frameworks sub-corpora were Structural and Multidisciplinary Optimization, Journal of Computational Physics, Environmental Modeling & Software, and Journal of Hydrology (see Fig. 8). The 10 most cited papers from across these top five journals came from Environmental Modeling & Software (2), Structural and Multidisciplinary Optimization (3), Journal of Hydrology (2), Journal of Computational Physics (1), and Computer Methods in Applied Mechanics and Engineering (2), and are detailed in Table 5 under Supplementary Material.

The top-cited "framework" related papers from these journals (Table 7) showcase a range of issues but particularly address the lack of uniformity in the UA/SA approaches used in their respective fields. These fields include:

- environmental modeling – evaluating performance (Bennett et al., 2013), improving confidence in model outcomes, and handling uncertainty (Bennett et al., 2013; Kuczera et al., 2006; Refsgaard et al., 2007), UA for hydrological (SWAT) models (Yang et al., 2008),
- optimization – topology optimization (Sigmund and Maute, 2013), Finite Element Methods (Blatman and Sudret, 2011; Moens and Vandepitte, 2005), level set methods for structural topology optimization (van Dijk et al., 2013), high-dimensional computationally expensive black-box problems (Shan and Wang, 2010), and
- scientific computing - handling uncertainty (Roy and Oberkampf, 2011).

Outlines of procedures, guidelines, comparisons of methods, and suggestions for future research resolve the issues raised in these papers. These papers are highly-cited, indicating that they have had an impact on the research community, at least within their respective fields. It should be noted here that existence of highly cited papers itself does not indicate widespread application of suggested good or best practice and should not be taken as evidence. The review conducted by Saltelli et al. (2019) concludes that there is a "worrying lack of standards and good practices", although it is acknowledged that the review focuses on older papers and may not capture recent trends. Certainly awareness appears to have increased, if not adoption of practices.

Similarly, a keyword search for "best practices" identified 132 papers across the surveyed period. By "best practices" we refer to practices in modeling and uncertainty management that promote transparency and reliability of results. The high citation counts of papers relating to frameworks (Table 7) and the growth in best practices publications in absolute terms (Fig. 9) suggest increasing interest in uncertainty management, particularly improving the reliability and effectiveness of UA/SA. Whether the modelers take up the suggestions in these papers is yet to be seen. Modelers can be encouraged to follow guidelines for reliable and effective treatment of UA/SA if the available software implementing UA/SA is designed in accordance with these guidelines (and if modelers

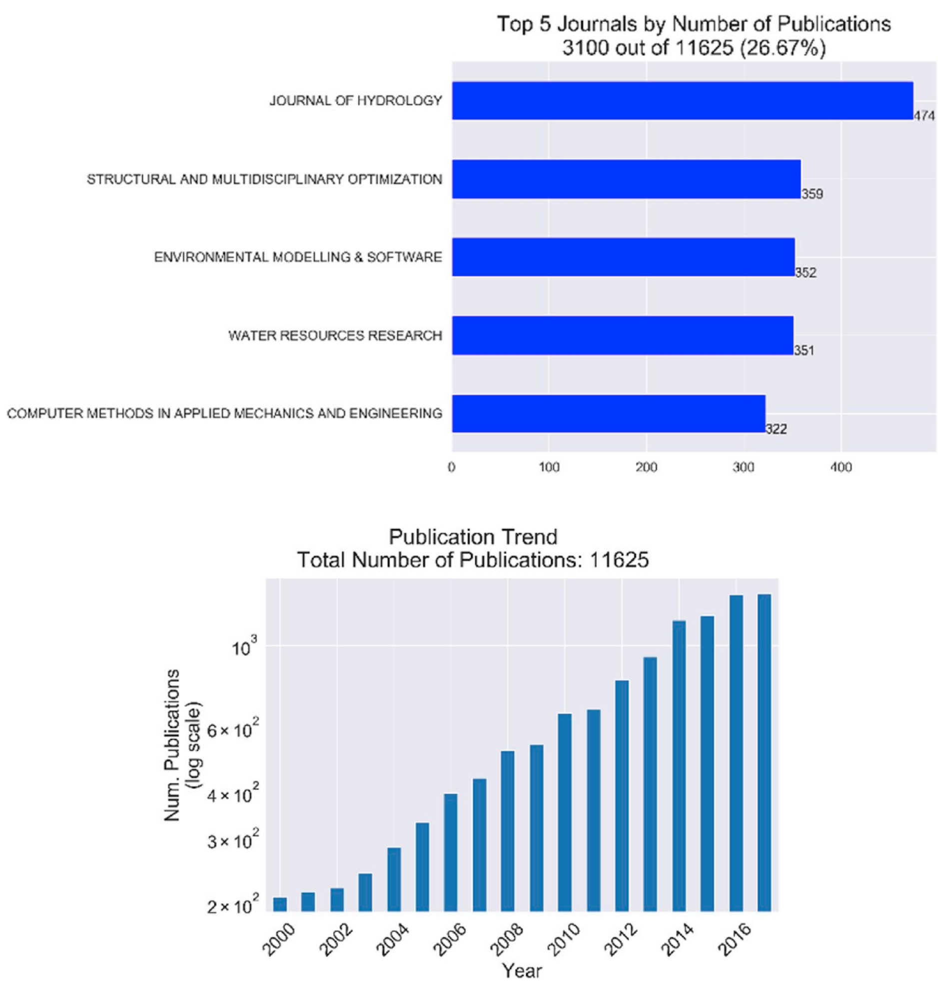


Fig. 5. Publication trends over 2000 to 2017. Journal of Hydrology contributed the most publications in the time frame (474, see top panel). Publications within the field have been occurring at an exponential rate (bottom panel).

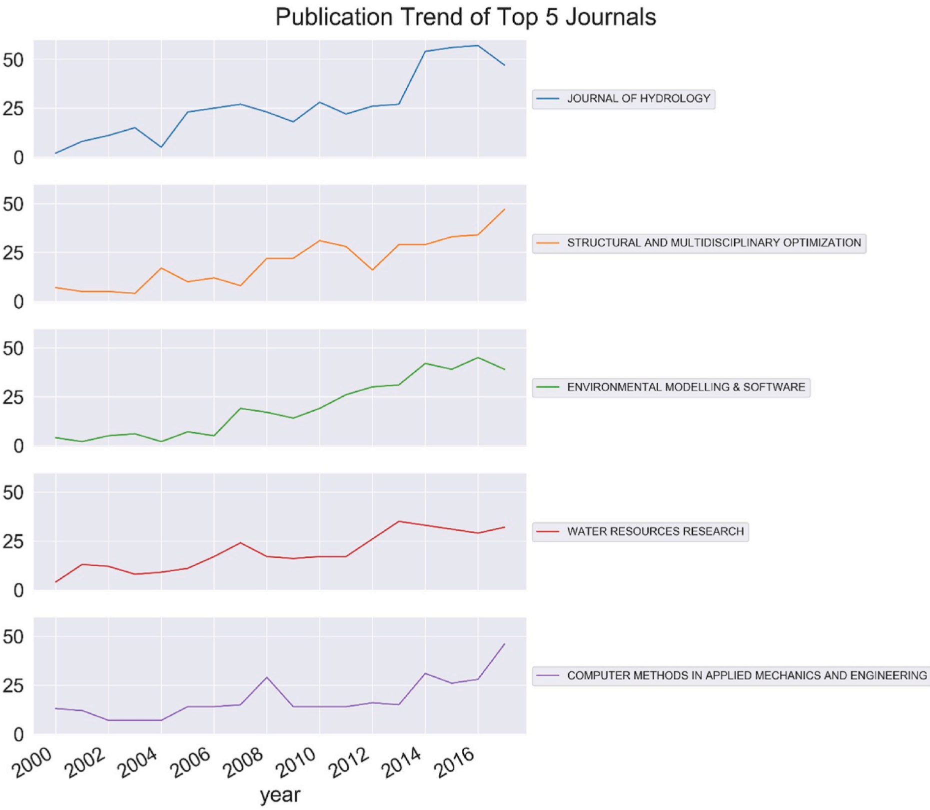
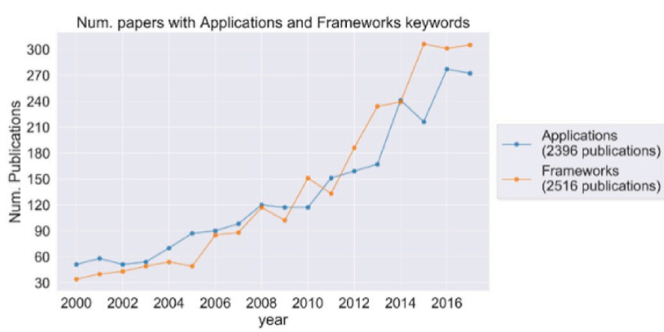


Fig. 6. Publication trend by journal across the timeframe. All journals in the top 5 (by number of publications) saw an increase in publications related to the keywords used.



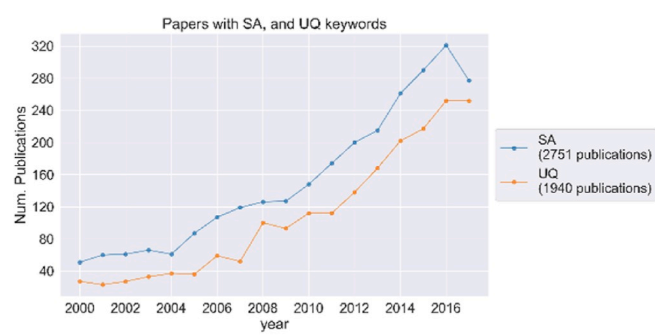


Fig. 7. Publication trends of papers relating to application of UA/SA and frameworks (left) as well as those related to sensitivity analysis, and uncertainty analysis and quantification (right). While publications are increasing in absolute terms, relative to their respective corpora, works on uncertainty frameworks are increasing while SA related papers have decreased, indicating a shift in focus.

make use of such software).

4.4.2. *Recent developments*

Recent impactful publications in sensitivity analysis suggest a shift away from local sensitivity methods. Prior to 2010, 'one-factor-at-a-time' (OAT) local SA was the most prevalent practice in the literature (Saltelli and Annoni, 2010) with a later revisit indicating that while this was still the case for papers published in Science and Nature, GSA methods were gaining traction (Ferretti et al., 2016). A more recent bibliometric review conducted by Saltelli et al. (2019) comes to a similar conclusion across 19 subject areas in which modeling features heavily, although the growth of OAT-related publications is shown to significantly out-pace GSA related publications. Within the presented corpora publications with OAT related keywords do decrease slightly over the past two decades (down roughly 1% compared to the entire corpora), with an uptick in the absolute number of publications post-2010 (see Fig. 10).

Although OAT is said to be a common method (see for example Shin et al., 2013) it may not have featured heavily prior to 2010 due to 1) researchers not reporting OAT use, 2) modelers using custom implementations of OAT, and 3) the software surveyed in our analysis did not support OAT, which discourages modelers from using this method (i.e. they select from available methods). Analysis conducted here indicates an increase in reported GSA keywords post-2010 (Fig. 11) – after the publication of "How to avoid a perfunctory sensitivity analysis" (Saltelli and Annoni, 2010). This paper was identified as a highly cited publication in the initial corpora (Table 6 in the supplementary material), a key contribution being the demonstrated inefficacy of OAT analyses using a geometric proof. The uptick in publications with the OAT-related keywords appears to correlate with the number of papers citing the paper by Saltelli and Annoni (2010), shown in Fig. 12. This may contribute to the rise in publications with OAT related keywords in the

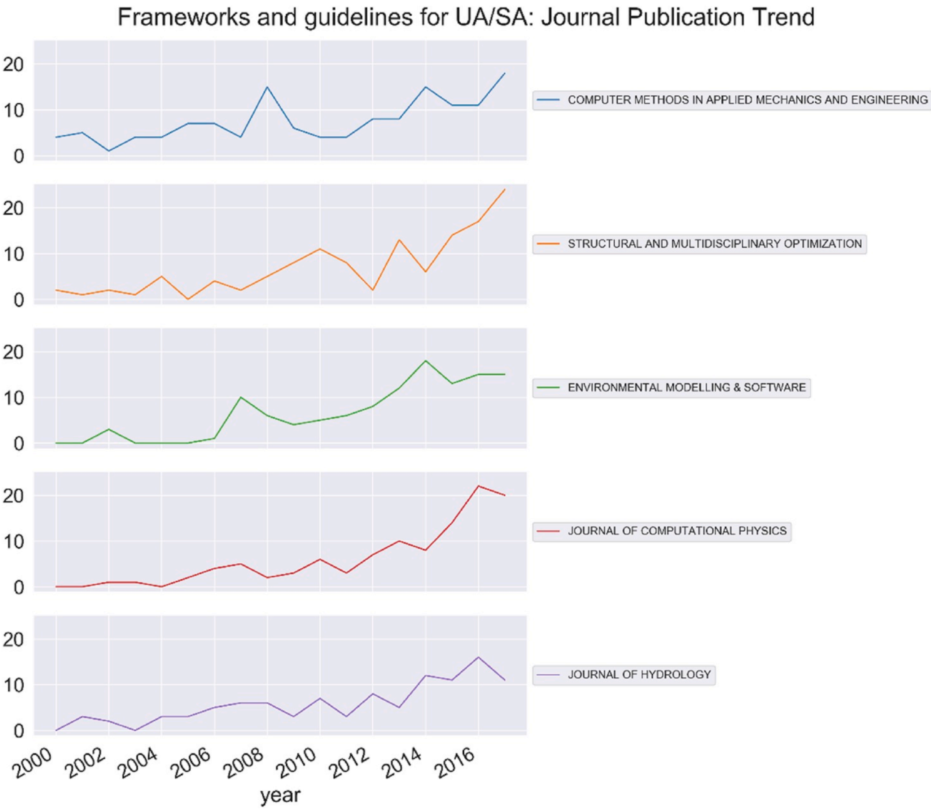


Fig. 8. The top five active journals publishing papers related to UA/SA frameworks. All journals have an increasing publication trend over the given timeframe.

corpora and as identified by Saltelli et al. (2019). The detected increase in GSA papers may reflect the start of changing attitudes towards SA in recognition of the importance of global sensitivity analyses. Increased awareness in the past decade has led to the use and development of more efficient and comprehensive UA/SA techniques and approaches.

Improved approaches put forth in the past decade attempt to enhance the computational efficiency of generating a global sensitivity measure (or range of measures as the case may be) from a single sample set, itself said to be more representative of the possible parameter space (e.g. Razavi et al., 2019). In particular there has been a renewed interest in GSA based on (statistical) design of experiment approaches, as these methods are capable of producing global sensitivity measures at an acceptable computational cost (Gan et al., 2014; Saltelli, 2017). Such approaches refer to methods that utilize a deterministic sample set, for example the aforementioned Sobol', Latin Hypercube, and Morris methods (Saltelli, 2017).

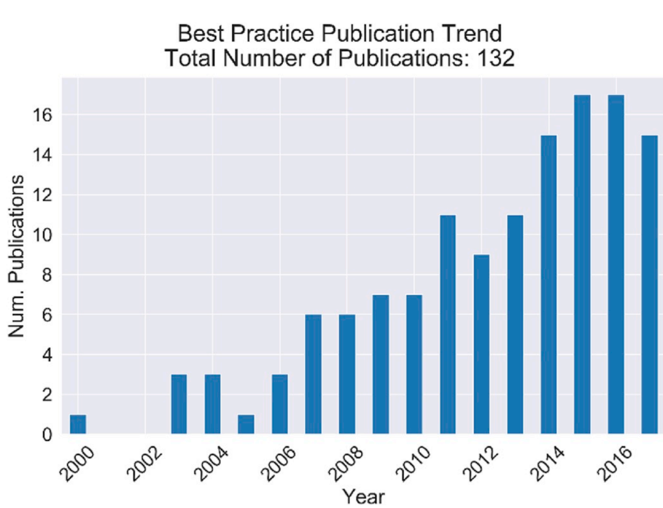


Fig. 9. Publication trend of papers with keywords relating to best practices. Notice the larger volume of publications in the years 2014–2017.

Despite the increased interest in GSA evidenced by the bibliometric analysis, local SA and OAT methods are still in widespread use, if any SA is conducted at all. Shin et al. (2013) for example, found that only 7% (11 of 164) of papers surveyed conducted any SA, of which five applied OAT. It is difficult to ascertain the full extent of OAT analysis through keyword analysis, as researchers applying this technique may not make explicit reference to this form of analysis. Possible reasons for the relatively slow uptake of GSA methods are listed in Ferretti et al. (2016), including perceived complexity in the application of GSA. Modelers were characterized as being hesitant due to a lack of experience with GSA methods. We also find in the literature a prevalence of self-implemented UA/SA; that is, modelers using their own code in place of existing and often open-source software tools. Not using, or otherwise contributing to, readily available, widely used, and well-tested software represents a duplication of work. This can be somewhat alleviated by greater awareness of and access to the available software tools that simplify the application and use of such analyses. Those developing tools and methods, for their part, could strive to improve ease of use and lower the technical and conceptual barriers to uptake of their software.

Pianosi et al. (2016) outline three principles of good practice for a sensitivity analysis package: 1) the ability to apply multiple sensitivity analyses to one sample, 2) provision of tools to assess and revise user choices, and 3) inclusion of visualization tools. Regarding point 1, early software releases tended to be platform, method, or model specific (see Table 4 for specific examples). In recent years the available software has been made for more general-purpose use, offering a more comprehensive approach to UA/SA with multiple methods supported. The lack of collaborative development is also reportedly an issue, with researchers preferring to develop their own toolset and as a consequence siloing advances, at least in the short to medium term. Usability, especially for

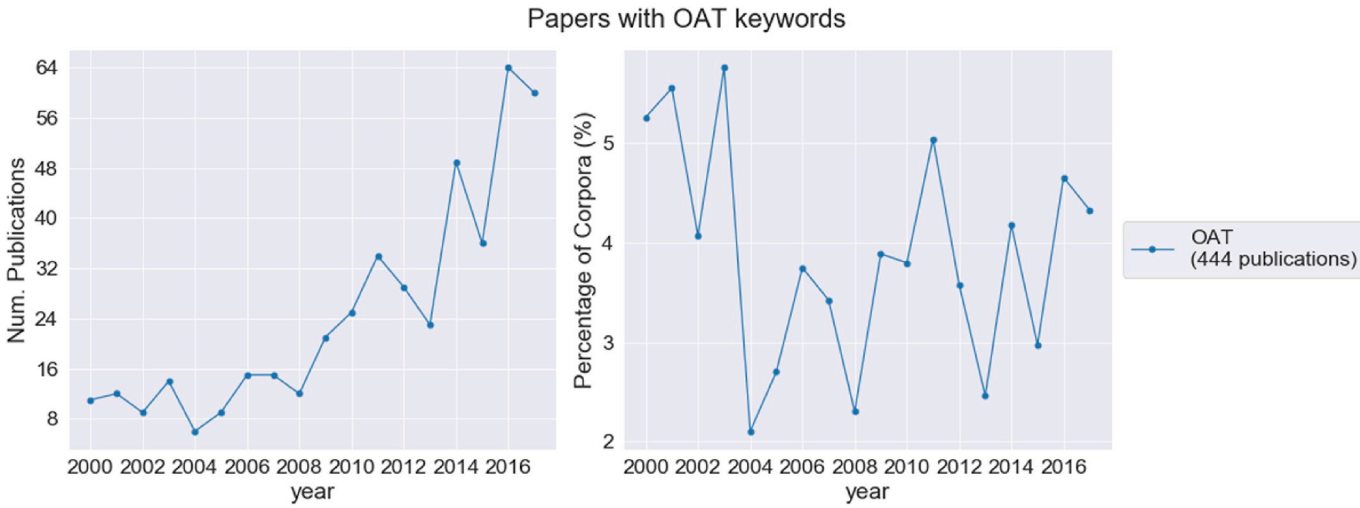


Fig. 10. Absolute and relative trends of publications with OAT keywords. Although publications increase in absolute terms, relative to the corpora yearly publications with OAT keywords decrease over the timeframe.

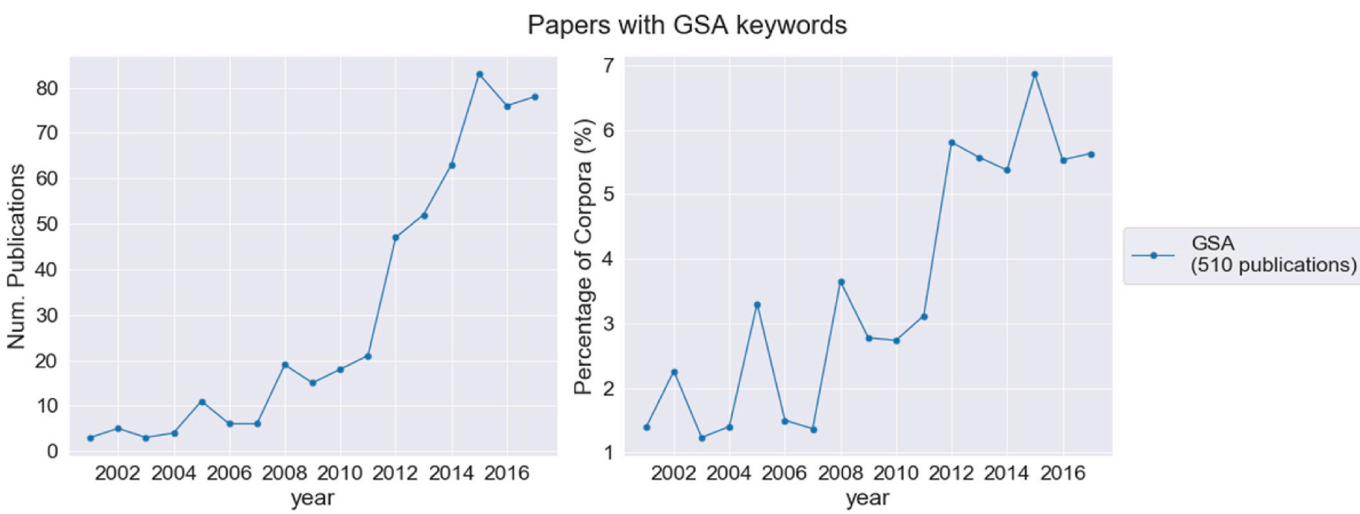


Fig. 11. Absolute and relative trends of publications with GSA keywords. Publications with GSA keywords increase over the timeframe both in absolute and relative terms.

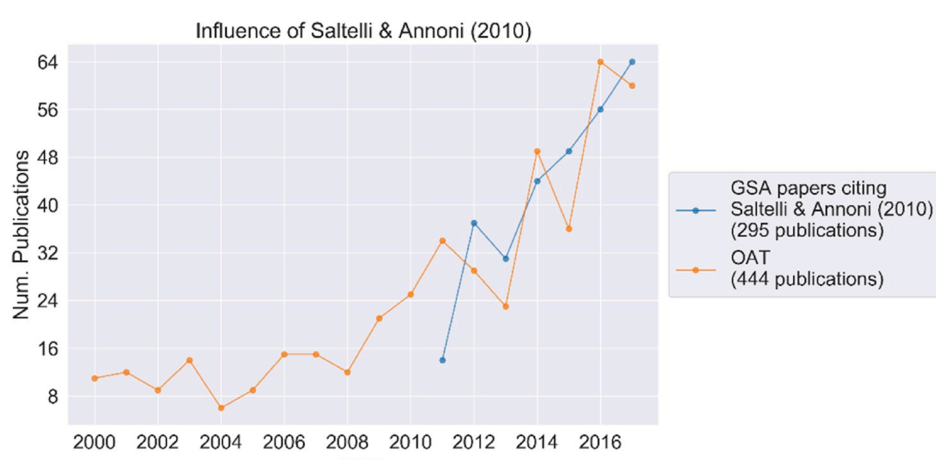


Fig. 12. Papers citing Saltelli and Annoni (2010) appear to be driving the uptick in publications with OAT-related keywords post-2010, possibly as authors give their reason(s) for not relying on OAT.

novices, is an ongoing concern. While many sampling and analysis approaches are amenable to cross-use (e.g. a mix-and-match approach) there is often no limitation in the application of methods (within the packages) which safeguards a user against inappropriate and incompatible mixes, e.g. Sobol' analysis on a Latin Hypercube sample.

Efforts to address these issues and criticisms are evident in the various communities, however, with later packages often offering detailed documentation including usage examples and tutorials (see previous section). Well-known test functions, such as the Ishigami function (Ishigami and Homma, 1990), Sobol' G-Function (Saltelli and Sobol, 1995), the example Lake Problem (Hadka et al., 2015) as well as case studies for research and educational purposes, are often included.

PEST (Doherty, 2018), for example, provides a UA tutorial including two worked examples of hydrological models. Another example is SAFE (Pianosi et al., 2015), which, by providing commented code in workflow scripts, allows beginner users to implement UA/SA more easily and advanced users to improve their methodology. The packages in our survey did not appear to provide guidance through UA/SA theory outside of extensive reading lists, although it is acknowledged that this may be out of scope for those maintaining the tool. A lack of guidance as such may hinder uptake by practitioners of both the software and GSA in general. Generally, the available packages still operate on an understanding that users have appropriate background knowledge of or experience with UA/SA. Many of the packages identified in this review appear to have been abandoned: this perhaps indicates the importance of an active user community to share knowledge and update code.

The corpora give evidence to newer methods that are in development and reflects a continued interest in improving UA/SA. Examples of more recently developed techniques are VARS and active subspaces (developed in 2016 and 2011 respectively). VARS (Variogram Analysis of Response Surfaces) uses variograms as a measure of sensitivity. A variogram is a function describing the spatial dependence of, in the case of SA, the parameter space, that is, how much the variance in parameter values is dependent on the distance between parameters in parameter space. Variogram-related publications began in 2001,however, those specifically relating to the application of variograms to SA only appear from 2016. There are five publications relevant to variogram-based SA in the corpora, totaling 74 citations and the highest citation average is 13 (Razavi and Gupta, 2016).

The top-cited variogram paper (Razavi and Gupta, 2016) presents the method as a linkage between existing derivative- and variance-based GSA methods and demonstrates that the approach reduces computational cost. Over approximately 20 000 and 100 000 model runs, the VARS sensitivity estimates had less uncertainty than Sobol and Morris indices. These relatively new methods, though currently lacking citations, do appear to be methods with development potential due to, for example, current user interest and improvements to the efficiency and comprehensibility of UA/SA methods.

Active subspaces is a dimension reduction technique that identifies directions in parameter space that have a greater influence on the model output. These directions are described as being "active" and their identification aids in reducing the dimensionality of a model by avoiding perturbations across inactive areas of parameter space, thereby reducing computational cost (Constantine et al., 2015). Through this method parameters of importance and their rankings can be obtained (Jefferson et al., 2015). Papers relating to active subspaces first appear in 2015, there are eight in total in the corpora. Citation analysis does not indicate particularly that this new method is being taken up quickly, total citations for all publications was 83, and the publication with the highest citation average had an average of 7.67 (Constantine et al., 2015). The top-cited active subspaces paper (Constantine et al., 2015) details an application of the method to numerical simulation, and an implementation may be found in the 'Effective Quadratures' package for Python (Seshadri and Parks, 2017)

Another technique of interest is HDMR (High Dimensional Model Reduction): the companion paper (Ziehn and Tomlin, 2009) for the method and supporting software came through as a highly cited publication in this analysis (see Table 7). HDMR is an emulation method that improves variance-based SA methods, such as the Sobol' method. Citing articles for Ziehn and Tomlin (2009) continue up to 2019 (identified through manual processes). In fact, 7 of the 32 returned publications in the corpora were published in 2017, indicating a continued interest in the method. In the corpora, the publication with the most citations has 158 (Alış and Rabitz, 2001) and the highest citation average is 14 (Ziehn and Tomlin, 2009).

Furthermore, alternative methods for handling uncertainty have been developed, especially to handle scenarios in which there is large uncertainty, but in which accurate predictions are necessary for future policy making. Software for these alternate methods is deemed out-of-scope for this study but for completeness sake, one such proposed approach is Exploratory Modeling and Analysis. Rather than simply minimizing uncertainty in an attempt to produce an accurate or precise prediction, uncertainty is treated as inevitable. Decision making processes are guided through the exploration of possible outcomes generated through computational experiments and responses planned (Eker et al., 2018; Kwakkel and Pruyt, 2013).

5. Limitations

The bibliometric analysis presented here is limited by the scope of the WoS database, the specific search terms used, the initial time frame and the included fields of study (with the analysis focused on applications in environmental modeling). Search query results may also differ over time due to indexing artefacts with implications for the resulting trend and citation analysis. A bias towards open-source software literature may be perceived as these were the easiest to analyze. That said, it is not claimed that the analysis conducted herein uncovered all software packages currently in use or the full extent to which they are being used.

A known issue is the lack of attributions, citations, and reporting of software used for research, making it difficult to find their mention, especially when the analysis relied on abstract text. Other software may not be referenced simply because their use is taken to be a fundamental part of the (programming) language ecosystem, for example, the R 'sensitivity' package or 'sci-kit learn' (for Python). It was also difficult to search within the corpora for packages with names common to other applications (taking as a particularly difficult example, the R 'sensitivity' package).

In our own process of sorting the generated database, decisions whilst manually sorting and choosing the software collection papers were subject to inherent bias – although this process was kept as transparent and objective as possible (see Notebook 5a "Finding software packages by keyphrase extraction"). Another process which limited the generality of our findings was that of refining the search terms and results. Limiting the scope of the results was necessary to facilitate analysis of the most relevant publications. Iterative use of the topic model achieved this, however, it is entirely possible that relevant

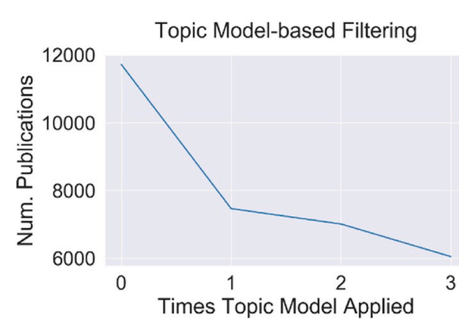


Fig. 13. Plot of the topic model filtering. Note the decrease in publications with each application, which aided in limiting the scope of results but also risked removing relevant publications.

publications will have been removed (Fig. 13). Of particular note is the possible under-representation of articles on emulators and surrogate modeling within the Software corpora. Omitted publications were assumed to be irrelevant or that relevant issues were captured by the papers that remained in the desired corpora. For more information, see Notebook 4 "UASA Topic modeling".

6. Conclusion and future directions

The analysis presented here indicates that UA considerations are increasingly included in the published literature with a slight decrease in the reported use of OAT methods. The identified literature reflects greater attention paid to guidelines for the use of UA/SA over the past decade, itself perhaps indicating advances in the application of UA/SA. Greater interest in the use of UA/SA for rigorous model testing is apparent, although whether modelers embrace and adopt the suggested guidelines towards the treatment, assessment and analysis of UA/SA (e.g. as discussed in Eker et al., 2018; Saltelli et al., 2019) remains to be seen.

The literature also suggests that a wide variety of software has become available in the past two decades, aimed at both non-programmatic audiences and for specific programming languages. The majority of these identified software packages does not support local OAT analyses, which may indicate a general move away from depending on local SA. More recently developed software packages that implement multiple methods with open source code and documentation, with little restriction (in terms of software licencing) to the end-user, are becoming the prevalent distribution format.

While there is a variety of software tools available, the trend of publications on UA/SA tooling has remained largely flat. This trend may be due to the relative infancy of the available tools, or a perceived complexity in their application. For one, while many of the surveyed software provide usage examples and documentation, their use typically assumes 1) experience with the underlying programming language, or 2) intimate familiarity with the methods provided, their pros and cons and contextual suitability. Little guidance is available, aside from extensive reading lists.

The indicated lack of uptake in this analysis may also be because software-specific publications have been largely filtered out from the corpora. These relevant publications may be concentrated within conference proceedings (which were removed from the corpora) or other topic areas not included in the initial publication search. Publications that are application/method focused may not explicitly mention the software used in the abstract. For these reasons it is difficult to concretely conclude whether those involved in environmental modeling are embracing the available UA/SA software tools or if custom "home-grown" solutions are preferred, itself indicating perhaps a lack of awareness of the available software packages.

That said, usability and user-friendliness were found to be a general issue. Users are expected to be adept and experienced enough to produce and interpret results themselves. Even in cases where visualization processes are provided, users may require a different approach for their analyses. In the case of novices, interpreting provided analyses requires first understanding a body of work usually provided in the form of a (often large) reading list of relevant papers. This may explain, in part, a preference for custom "home-grown" solutions where the developers write tools specific to their needs to avoid "adoption cost"; time needed to learn how to use an existing tool effectively. The complexity of existing tools, real or perceived, may contribute to the issue of (lack of) uptake. In cases where the perceived cost of adoption is high, the prospective user may find it easier to apply OAT or otherwise implement their own custom solution to perform common UA/SA methods, which amounts to duplication of effort across the scientific community.

This then raises the question of what constitutes a 'thorough' UA/SA package. In this survey, the most comprehensive software (R sensitivity, Simlab, and SALib) provide users with the widest assortment of UA/SA methods with (limited) visualization capability and test functions. These target languages prevalent in the sciences (R, Matlab, and Python respectively) that are supported by an active community, which may explain their longevity and/or popularity. The prevalence of open-source, community-led efforts evident in more recent software tools suggests that an open development culture is a prerequisite to wide-spread adoption – perhaps an unremarkable observation due to the scientific context and focus of UA/SA research.

Developers and maintainers of UA/SA tools could support and encourage wider application of GSA processes by moving towards 1) an open development process, 2) placing further attention on expanding documentation, preferably in an easily digestible form, and 3) improving usage guidelines and promoting user-centric interfaces and workflows.

Point 1 is to encourage the sharing of knowledge and experience across the disciplines that rely on modeling, to leverage expertise and experience globally rather than siloing advances. On point 2, UA/SA software developers could further leverage the open-collaboration model and (re-)use explanations and examples from one another. Examples of both simple and complex workflows could be given (e.g. in a "cookbook" or "recipe" documentation style). Point 3 should not be taken to mean that all packages should provide a GUI. Rather, general-purpose UA/SA tools should have processes in place to prevent or limit unintentional or ill-informed analyses from occurring. A particular pain point is the ability to mix-and-match sampling and analysis methods regardless of whether it makes sense to do so.

While UA/SA tools have largely addressed the three steps defined by Pianosi et al. (2015) (sample parameter space, run model, analyze results), the workflow - that is implicit or explicit steps in the use and application of the software - could be improved so that modelers are able to move from each step without issue. Recently developed packages indicate that such improvements to the workflow are being made, with attention to usability, open-source code, and tools for analyzing results. Researchers and modelers, particularly those new to UA/SA, need software designed with usability in mind. It is expected that such software will support UA/SA in more areas and encourage rigorous and reliable UA/SA, which will in turn allow for more informed decision-making.

Software availability

Code and representative data used for this analysis can be found at https://github.com/frog7/uasa-trends (10.5281/zenodo.3406946).

Software used to support analysis can be found at https://github.com/ConnectedSystems/wosis (10.5281/zenodo.3406947).

Declaration of competing interest

The second author has contributed usability and performance improvements to the SALib Python library. All other authors declare no potential sources of conflict.

Acknowledgements

The corresponding author (second in the author list) is supported through an Australian Government Research Training Program (AGRTP) Scholarship, and a top-up scholarship from the Hilda-John Endowment Fund.

The authors would like to thank and acknowledge Joseph Guillaume, Barbara Robson, and the anonymous reviewers for their highly valued comments and suggestions. The use of work by Titipat Achakulvisut (author of the 'wos_parser' Python package, https://github.com/titipata/wos_parser) and Enrico Bacis (author of the 'WoS Client' package, https://github.com/enricobacis/wos) is acknowledged. We would also thank and acknowledge Clarivate Analytics for providing access and use of the Web of Science database, and to their staff for providing clarifications to technical details of the available API without which this work would not have been possible.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2019.104588.

References

Achakulvisut, T., Acuna, D.E., Ruangrong, T., Kording, K., 2016. Science concierge: a fast content-based recommendation system for scientific publications. PLoS One 11, e0158423. https://doi.org/10.1371/journal.pone.0158423.

Adams, B.M., Ebeida, M.S., Eldred, M.S., Gianluca, G., Jakeman, J.D., Maupin, K.A., Monschke, J.A., Adam Stephens, J., Swiler, L.P., Vigil, D.M., Wildley, T.M., Bohnhoff, W.J., Dalbey, K.R., Eddy, J.P., Frye, J.R., Hooper, R.W., Hu, K.T., Hough, P.D., Khalil, M., Ridgway, E.M., Winokur, J.G., Rushdi, A., 2010. Dakota : a Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. Version 5.0, User's Manual. (No. SAND2010-2183, 991842). https://doi.org/10.2172/991842.

Alış, Ö.F., Rabitz, H., 2001. Efficient implementation of high dimensional model representations. J. Math. Chem. 29, 127–142. https://doi.org/10.1023/A:1010979129659.

Arora, S., Ge, R., Moitra, A., 2012. Learning Topic Models - Going beyond SVD.

Ascough II, J.C., Fischer, C., Lighthart, N.P., David, O., Green, T.R., Kralisch, S., 2015. The model optimization, uncertainty, and sensitivity analysis (MOUSE) Toolbox: overview and application. In: Proc. Hydrology Days. Presented at the Proc. Hydrology Days 2015. Colorado State University, Fort Collins, Colorado, pp. 17–28.

Becker, W.E., Tarantola, S., Deman, G., 2018. Sensitivity analysis approaches to high-dimensional screening problems at low sample size. J. Stat. Comput. Simul. 88, 2089–2110. https://doi.org/10.1080/00949655.2018.1450876.

Beel, J., Gipp, B., Langer, S., Breitinger, C., 2016. Research-paper recommender systems: a literature survey. Int. J. Digit. Libr. 17, 305–338. https://doi.org/10.1007/s00799-015-0156-0.

Belyaev, M., Burnaev, E., Kapushev, E., Panov, M., Prikhodko, P., Vetrov, D., Yarotsky, D., 2016. GTApprox: surrogate modeling for industrial design. Adv. Eng. Software 102, 29–39. https://doi.org/10.1016/j.advengsoft.2016.09.001.

Bennett, N.D., Croke, B.F.W., Guariso, G., Guillaume, J.H.A., Hamilton, S.H., Jakeman, A.J., Marsili-Libelli, S., Newham, L.T.H., Norton, J.P., Perrin, C., Pierce, S. A., Robson, B., Seppelt, R., Voinov, A.A., Fath, B.D., Andreassian, V., 2013. Characterising performance of environmental models. Environ. Model. Softw 40, 1–20.

Beven, K., Binley, A., 1992. The future of distributed models: model calibration and uncertainty prediction. Hydrol. Process. 6, 279–298. https://doi.org/10.1002/hyp.3360060305.

Bezanson, J., Edelman, A., Karpinski, S., Shah, V., 2017. Julia: a fresh approach to numerical computing. SIAM Rev. 59, 65–98. https://doi.org/10.1137/141000671.

Blatman, G., Sudret, B., 2011. Adaptive sparse polynomial chaos expansion based on least angle regression. J. Comput. Phys. 230, 2345–2367. https://doi.org/10.1016/j.jcp.2010.12.021.

Bornmann, L., Mutz, R., 2015. Growth rates of modern science: a bibliometric analysis based on the number of publications and cited references. J. Assoc. Inf. Sci. Technol. 66, 2215–2222. https://doi.org/10.1002/asi.23329.

Campolongo, F., Saltelli, A., Cariboni, J., 2011. From screening to quantitative sensitivity analysis. A unified approach. Comput. Phys. Commun. 182, 978–988. https://doi.org/10.1016/j.cpc.2010.12.039.

Castaings, W., Borgonovo, E., Morris, M.D., Tarantola, S., 2012. Sampling strategies in density-based sensitivity analysis. Environ. Model. Softw 38, 13–26. https://doi.org/10.1016/j.envsoft.2012.04.017.

Chen, Y., Zhang, H., Liu, R., Ye, Z., Lin, J., 2019. Experimental explorations on short text topic mining between LDA and NMF based Schemes. Knowl. Based Syst. 163, 1–13. https://doi.org/10.1016/j.knosys.2018.08.011.

Constantine, P.G., Emory, M., Larsson, J., Iaccarino, G., 2015. Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot II scramjet. J. Comput. Phys. 302, 1–20. https://doi.org/10.1016/j.jcp.2015.09.001.

Crestaux, T., Le Maitre, O., Martinez, J.-M., 2009. Polynomial chaos expansion for sensitivity analysis. Reliab. Eng. Syst. Saf. 94, 1161–1172. https://doi.org/10.1016/j.ress.2008.10.008.

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, 3873–3878. https://doi.org/10.1063/1.1680571.

Czitrom, V., 1999. One-Factor-at-a-Time versus designed experiments. Am. Stat. 53, 126–131. https://doi.org/10.1080/00031305.1999.10474445.

Doherty, J., 2018. Model-Independent Parameter Estimation User Manual Part I: PEST, SENSAN and Global Optimisers.

Douglas-Smith, D., Iwanaga, T., 2019. UASA Trends.

Eker, S., Rovenskaya, E., Obersteiner, M., Langan, S., 2018. Practice and perspectives in the validation of resource management models. Nat. Commun. 9, 1–10. https://doi.org/10.1038/s41467-018-07811-9.

Fedra, K., 1983. Environmental Modeling under Uncertainty: Monte Carlo Simulation [WWW Document]. URL. http://pure.iiasa.ac.at/id/eprint/2152/ (accessed 4.10.19).

Feinberg, J., Langtangen, H.P., 2015. Chaospy: an open source tool for designing methods of uncertainty quantification. J. Comput. Sci. 11, 46–57. https://doi.org/10.1016/j.jocs.2015.08.008.

Ferretti, F., Saltelli, A., Tarantola, S., 2016. Trends in sensitivity analysis practice in the last decade. Sci. Total Environ. 568, 666–670. https://doi.org/10.1016/j.scitotenv.2016.02.133.

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, 269–285. https://doi.org/10.1016/j.envsoft.2013.09.031.

Guillaume, J.H.A., Jakeman, J.D., Marsili-Libelli, S., Asher, M., Brunner, P., Croke, B., Hill, M.C., Jakeman, A.J., Keesman, K.J., Razavi, S., Stigter, J.D., 2019. Introductory overview of identifiability analysis: A guide to evaluating whether you have the right type of data for your modeling purpose. Environmental Modelling & Software 119, 418–432. https://doi.org/10.1016/j.envsoft.2019.07.007.

Haddaway, N.R., Westgate, M.J., 2018. Predicting the time needed for environmental systematic reviews and systematic maps. Conserv. Biol. 0 https://doi.org/10.1111/cobi.13231.

Hadka, D., Herman, J., Reed, P., Keller, K., 2015. An open source framework for many-objective robust decision making. Environ. Model. Softw 74, 114–129. https://doi.org/10.1016/j.envsoft.2015.07.014.

Helton, J.C., 1993. Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal. Reliab. Eng. Syst. Saf. 42, 327–367. https://doi.org/10.1016/0951-8320(93)90097-I.

Herman, J., Usher, W., 2017. SALib: an open-source Python library for Sensitivity Analysis [WWW Document] J. Open Source Softw. 2 (9), 97. https://doi.org/10.21105/joss.00097.

Hough, B., Fu, C., Paliwal, S., 2016. savvy: visualize high dimensionality sensitivity analysis data. Updated with full sensitivity analysis from ligpy model. Zenodo. https://doi.org/10.5281/zenodo.53099.

Hunt, M., Haley, B., McLennan, M., Koslowski, M., Murthy, J., Strachan, A., 2015. PUQ: a code for non-intrusive uncertainty propagation in computer simulations. Comput. Phys. Commun. 194, 97–107. https://doi.org/10.1016/j.cpc.2015.04.011.

Iman, R.L., Helton, J.C., 1988. An investigation of uncertainty and sensitivity analysis techniques for computer models. Risk Anal. 71–90. https://doi.org/10.1111/j.1539-6924.1988.tb01155.x.

Iooss, B., Pujol, A.J., Boumhaout, G., with contributions from K, Veiga, S.D., Delage, T., Fruth, J., Gilquin, L., Guillaume, J., Gratiet, L.L., Lemaitre, P., Nelson, B.L., Monari, F., Oomen, R., Rakovec, O., Ramos, B., Roustant, O., Song, E., Staum, J., Sueur, R., Touati, T., Weber, F., 2018. Sensitivity: Global Sensitivity Analysis of Model Outputs.

Ishigami, T., Homma, T., 1990. An importance quantification technique in uncertainty analysis for computer models. In: [1990] Proceedings. First International Symposium on Uncertainty Modeling and Analysis. Presented at the [1990] Proceedings. First International Symposium on Uncertainty Modeling and Analysis, pp. 398–403. https://doi.org/10.1109/ISUMA.1990.151285.

Iwanaga, T., Douglas-Smith, D., 2019. Wosis: Beta-Release. https://doi.org/10.5281/zenodo.3406947.

Jefferson, J.L., Gilbert, J.M., Constantine, P.G., Maxwell, R.M., 2015. Active subspaces for sensitivity analysis and dimension reduction of an integrated hydrologic model. Comput. Geosci. 83, 127–138. https://doi.org/10.1016/j.cageo.2015.07.001.

JRC, 2015. SIMLAB and Other Software - EU Science Hub - European Commission [WWW Document]. EU Science Hub. URL. https://ec.europa.eu/jrc/en/samo/simlab (accessed 12.5.18).

Kuczera, G., Kavetski, D., Franks, S., Thyer, M., 2006. Towards a Bayesian total error analysis of conceptual rainfall-runoff models: characterising model error using storm-dependent parameters. J. Hydrol. 331, 161–177. https://doi.org/10.1016/j.jhydrol.2006.05.010.

Kwakkel, J.H., Pruyt, E., 2013. Exploratory Modeling and Analysis, an approach for model-based foresight under deep uncertainty. Technol. Forecast. Soc. Chang. 80, 419–431. https://doi.org/10.1016/j.techfore.2012.10.005.

Leamer, E.E., 1985. Sensitivity analyses would help. Am. Econ. Rev. 75, 308–313.

Marelli, S., Sudret, B., 2014. UQLab: a framework for uncertainty quantification in Matlab. In: Vulnerability, Uncertainty, and Risk. Presented at the Second International Conference on Vulnerability and Risk Analysis and Management (ICVRAM) and the Sixth International Symposium on Uncertainty, Modeling, and Analysis (ISUMA). American Society of Civil Engineers, Liverpool, UK, pp. 2554–2563. https://doi.org/10.1061/9780784413609.257.

Matott, L.S., 2017. OSTRICH: an Optimization Software Tool, Documentation and User's Guide. Version 17.12.19 [WWW Document]. URL. http://www.eng.buffalo.edu/~lsmatott/Ostrich/OstrichMain.html (accessed 3.24.19).

Matott, L.S., Babendreier, J.E., Purucker, S.T., 2009. Evaluating uncertainty in integrated environmental models: a review of concepts and tools. Water Resour. Res. 45 https://doi.org/10.1029/2008WR007301.

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, 239–245. https://doi.org/10.2307/1268522.

Metropolis, N., Ulam, S., 1949. The Monte Carlo method. J. Am. Stat. Assoc. 44, 335–341. https://doi.org/10.2307/2280232.

Moens, D., Vandepitte, D., 2005. A survey of non-probabilistic uncertainty treatment in finite element analysis. Comput. Methods Appl. Mech. Eng. 194, 1527–1555. https://doi.org/10.1016/j.cma.2004.03.019.

Morris, M.D., 1991. Factorial sampling plans for preliminary computational experiments. Technometrics 33, 161–174. https://doi.org/10.2307/1269043.

Nakagawa, S., Samarasinghe, G., Haddaway, N.R., Westgate, M.J., O'Dea, R.E., Noble, D. W.A., Lagisz, M., 2018. Research weaving: visualizing the future of research

synthesis. Trends Ecol. Evol. 34 (3), 224–238. https://doi.org/10.1016/j.tree.2018.11.007.
Norton, J., 2015. An introduction to sensitivity assessment of simulation models. Environ. Model. Softw 69, 166–174. https://doi.org/10.1016/j.envsoft.2015.03.020.
Oakley, J., O'Hagan, A., 2004. Probabilistic sensitivity analysis of complex models: a Bayesian approach. J. R. Stat. Soc. 66, 751–769. https://doi.org/10.1111/j.1467-9868.2004.05304.x.
Oladyshkin, S., Nowak, W., 2012. Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. Reliab. Eng. Syst. Saf. 106, 179–190. https://doi.org/10.1016/j.ress.2012.05.002.
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.
Pianosi, F., Beven, K., Freer, J., Hall, J.W., Rougier, J., Stephenson, D.B., Wagener, T., 2016. Sensitivity analysis of environmental models: a systematic review with practical workflow. Environ. Model. Softw 79, 214–232. https://doi.org/10.1016/j.envsoft.2016.02.008.
Pianosi, F., Sarrazin, F., Wagener, T., 2015. A Matlab toolbox for global sensitivity analysis. Environ. Model. Softw 70, 80–85. https://doi.org/10.1016/j.envsoft.2015.04.009.
Poeter, E.P., Hill, M.C., 1999. UCODE, a computer code for universal inverse modeling1Code. Comput. Geosci. 25, 457–462. https://doi.org/10.1016/S0098-3004(98)00149-6 available at: http://water.usgs.gov/software/ground_ water.html1.
Rabby, G., Azad, S., Mahmud, Mufti, Zamli, K.Z., Mostafizur Rahman, M., 2018. A flexible keyphrase extraction technique for academic literature. In: Procedia Computer Science, the 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life, vol. 135, pp. 553–563. https://doi.org/10.1016/j.procs.2018.08.208.
Rakovec, O., Hill, M.C., Clark, M.P., Weerts, A.H., Teuling, A.J., Uijlenhoet, R., 2014. Distributed evaluation of local sensitivity analysis (DELSA), with application to hydrologic models. Water Resour. Res. 50, 409–426. https://doi.org/10.1002/2013WR014063.
Ratto, M., Pagano, A., 2010. Using recursive algorithms for the efficient identification of smoothing spline ANOVA models. ASA Adv. Stat. Anal. 94, 367–388. https://doi.org/10.1007/s10182-010-0148-8.
Razavi, S., Gupta, H.V., 2016. A new framework for comprehensive, robust, and efficient global sensitivity analysis: 1. Theory. Water Resour. Res. 52, 423–439. https://doi.org/10.1002/2015WR017559.
Razavi, S., Gupta, H.V., 2015. What do we mean by sensitivity analysis? The need for comprehensive characterization of "global" sensitivity in Earth and Environmental systems models: a Critical Look at Sensitivity Analysis. Water Resour. Res. 51, 3070–3092. https://doi.org/10.1002/2014WR016527.
Razavi, S., Sheikholeslami, R., Gupta, H.V., Haghnegahdar, A., 2019. VARS-TOOL: a toolbox for comprehensive, efficient, and robust sensitivity and uncertainty analysis. Environ. Model. Softw 112, 95–107. https://doi.org/10.1016/j.envsoft.2018.10.005.
Refsgaard, J.C., van der Sluijs, J.P., Hojberg, A.L., Vanrolleghem, P.A., 2007. Uncertainty in the environmental modelling process - a framework and guidance. Environ. Model. Softw 22, 1543–1556. https://doi.org/10.1016/j.envsoft.2007.02.004.
Regier, J., Fischer, K., Pamnany, K., Noack, A., Revels, J., Lam, M., Howard, S., Giordano, R., Schlegel, D., McAuliffe, J., Thomas, R., Prabhat, 2019. Cataloging the visible universe through Bayesian inference in Julia at petascale. J. Parallel Distrib. Comput. 127, 89–104. https://doi.org/10.1016/j.jpdc.2018.12.008.
Roos, M., Martins, T.G., Held, L., Rue, H., 2015. Sensitivity analysis for bayesian hierarchical models. Bayesian Anal. 10, 321–349. https://doi.org/10.1214/14-BA909.
Rose, S., Engel, D., Kogan, J., 2010. Automatic keyword extraction from individual documents. In: Text Mining: Applications and Theory. John Wiley & Sons, Ltd, pp. 1–20.
Roy, C.J., Oberkampf, W.L., 2011. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. Comput. Methods Appl. Mech. Eng. 200, 2131–2144. https://doi.org/10.1016/j.cma.2011.03.016.
Sagi, O., Rokach, L., 2018. Ensemble learning: a survey. Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 8, e1249. https://doi.org/10.1002/widm.1249.
Saltelli, A., 2017. Discussion Paper: Should Statistics Rescue Mathematical Modelling? arXiv:1712.06457 [stat].
Saltelli, A., Aleksankina, K., Becker, W., Fennell, P., Ferretti, F., Holst, N., Li, S., Wu, Q., 2019. Why so many published sensitivity analyses are false: a systematic review of sensitivity analysis practices. Environ. Model. Softw 114, 29–39.
Saltelli, A., Annoni, P., 2010. How to avoid a perfunctory sensitivity analysis. Environ. Model. Softw 25, 1508–1517. https://doi.org/10.1016/j.envsoft.2010.04.012.
Saltelli, A., Marivoet, J., 1990. Non-parametric statistics in sensitivity analysis for model output: a comparison of selected techniques. Reliab. Eng. Syst. Saf. 28, 229–253. https://doi.org/10.1016/0951-8320(90)90065-U.
Saltelli, A., Sobol, I.M., 1995. About the use of rank transformation in sensitivity analysis of model output. Reliab. Eng. Syst. Saf. 50, 225–239. https://doi.org/10.1016/0951-8320(95)00099-2.
Saltelli, A., Tarantola, S., 2002. On the relative importance of input factors in mathematical models: safety assessment for nuclear waste disposal. J. Am. Stat. Assoc. 97, 702–709.
Saltelli, A., Tarantola, S., Chan, K.P.S., 1999. A quantitative model-independent method for global sensitivity analysis of model output. Technometrics 41, 39–56.
Seshadri, P., Parks, G., 2017. Effective-Quadratures (EQ): Polynomials for Computational Engineering Studies. Journal of Open Source Software 2 (11), 166. https://doi.org/10.21105/joss.00166.
Shan, S., Wang, G.G., 2010. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Struct. Multidiscip. Optim. 41, 219–241. https://doi.org/10.1007/s00158-009-0420-2.
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. Journal of Hydrology 503, 135–152. https://doi.org/10.1016/j.jhydrol.2013.08.047.
Sigmund, O., Maute, K., 2013. Topology optimization approaches: a comparative review. Struct. Multidiscip. Optim. 48, 1031–1055. https://doi.org/10.1007/s00158-013-0978-6.
Sobol, I.M., 1993. Sensitivity analysis for non-linear mathematical models. Math. Model. Civ. Eng. 1, 407–414.
Sobol, I.M., Kucherenko, S., 2009. Derivative based global sensitivity measures and their link with global sensitivity indices. Math. Comput. Simulat. 79, 3009–3017. https://doi.org/10.1016/j.matcom.2009.01.023.
Spear, R.C., Hornberger, G.M., 1980. Eutrophication in peel inlet—II. Identification of critical uncertainties via generalized sensitivity analysis. Water Res. 14, 43–49.
Storlie, C., Helton, J., 2008. Multiple predictor smoothing methods for sensitivity analysis: description of techniques. Reliab. Eng. Syst. Saf. 93, 28–54. https://doi.org/10.1016/j.ress.2006.10.012.
Sun, X.Y., Newham, L.T.H., Croke, B.F.W., Norton, J.P., 2012. Three complementary methods for sensitivity analysis of a water quality model. Environ. Model. Softw 37, 19–29. https://doi.org/10.1016/j.envsoft.2012.04.010.
Tarantola, S., Gatelli, D., Mara, T.A., 2006. Random balance designs for the estimation of first order global sensitivity indices. Reliab. Eng. Syst. Saf. 91, 717–727. https://doi.org/10.1016/j.ress.2005.06.003.
van Dijk, N.P., Maute, K., Langelaar, M., van Keulen, F., 2013. Level-set methods for structural topology optimization: a review. Struct. Multidiscip. Optim. 48, 437–472. https://doi.org/10.1007/s00158-013-0912-y.
Vu-Bac, N., Lahmer, T., Zhuang, X., Nguyen-Thoi, T., Rabczuk, T., 2016. A software framework for probabilistic sensitivity analysis for computationally expensive models. Adv. Eng. Software 100, 19–31. https://doi.org/10.1016/j.advengsoft.2016.06.005.
Wagener, T., Kollat, J., 2007. Numerical and Visual Evaluation of Hydrological and Environmental Models Using the Monte Carlo Analysis Toolbox.
Wang, C., Duan, Q., Tong, C.H., Di, Z., Gong, W., 2016. A GUI platform for uncertainty quantification of complex dynamical models. Environ. Model. Softw 76, 1–12. https://doi.org/10.1016/j.envsoft.2015.11.004.
Wang, J., Li, X., Lu, L., Fang, F., 2013. Parameter sensitivity analysis of crop growth models based on the extended Fourier Amplitude Sensitivity Test method. Environ. Model. Softw 48, 171–182. https://doi.org/10.1016/j.envsoft.2013.06.007.
Westgate, M.J., Haddaway, N.R., Cheng, S.H., McIntosh, E.J., Marshall, C., Lindenmayer, D.B., 2018. Software support for environmental evidence synthesis. Nat. Ecol. Evol. (in press).
Westgate, M.J., Lindenmayer, D.B., 2017. The difficulties of systematic reviews. Conserv. Biol. 31, 1002–1007. https://doi.org/10.1111/cobi.12890.
Yang, J., Reichert, P., Abbaspour, K.C., Xia, J., Yang, H., 2008. Comparing uncertainty analysis techniques for a SWAT application to the Chaohe Basin in China. J. Hydrol. 358, 1–23. https://doi.org/10.1016/j.jhydrol.2008.05.012.
Young, P.C., Spear, R.C., Hornberger, G.M., 1978. Modeling badly defined systems: soem further thoughts. In: Proceedings SIMSIG Conference. Presented at the SIMSIG, Canberra, pp. 24–32.
Ziehn, T., Tomlin, A.S., 2009. GUI–HDMR – a software tool for global sensitivity analysis of complex models. Environ. Model. Softw 24, 775–785. https://doi.org/10.1016/j.envsoft.2008.12.002.