Environmental Modelling & Software 85 (2016) 217–228

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

A python framework for environmental model uncertainty analysis

Jeremy T. White [a, *], Michael N. Fienen [b], John E. Doherty [c]

[a] U.S. Geological Survey, Texas Water Science Center, 1505 Ferguson Lane, Austin, TX, USA
[b] U.S. Geological Survey, Wisconsin Water Science Center, 8505 Research Way, Middleton, WI, USA
[c] Watermark Numerical Computing, Brisbane, Australia

ARTICLE INFO

Article history:
Received 25 March 2016
Received in revised form
19 August 2016
Accepted 25 August 2016

Keywords:
Uncertainty analysis
FOSM
Python
Model independent
Highly-parameterized
Data worth analysis

ABSTRACT

We have developed pyEMU, a python framework for Environmental Modeling Uncertainty analyses, open-source tool that is non-intrusive, easy-to-use, computationally efficient, and scalable to highly-parameterized inverse problems. The framework implements several types of linear (first-order, second-moment (FOSM)) and non-linear uncertainty analyses. The FOSM-based analyses can also be completed prior to parameter estimation to help inform important modeling decisions, such as parameterization and objective function formulation. Complete workflows for several types of FOSM-based and non-linear analyses are documented in example notebooks implemented using Jupyter that are available in the online pyEMU repository. Example workflows include basic parameter and forecast analyses, data worth analyses, and error-variance analyses, as well as usage of parameter ensemble generation and management capabilities. These workflows document the necessary steps and provides insights into the results, with the goal of educating users not only in how to apply pyEMU, but also in the underlying theory of applied uncertainty quantification.

Published by Elsevier Ltd.

1. Introduction

Stakeholders and other consumers of environmental analysis are increasingly advocating for the quantification of uncertainty in the analysis provided to them (e.g. Hester and Coleman, 2014; Uusitalo et al., 2015). For example, environmental models are increasingly being used to inform the decisions related to resource management, which in this context, requires that forecasts made with such models be subjected to uncertainty analysis. Recent administrations have explicitly called for uncertainty analysis in forecasts. They specifically stated *"A good analysis provides … the results of formal sensitivity and other uncertainty analyses."* (Office of Management and Budget, 2003, p. 3). Further, *"If … uncertainty is reduced and accurately described, then decisions will be made that tend to make for better use of the resource and increase public benefits and/or reduce risk."*(National Science and Technology Council, 2007, p. 12). Recent environmental modeling analyses that include some form uncertainty treatment include Zhang et al. (2016); Zheng and Han (2016); Clough et al. (2016); Cho et al. (2016) among others. Many of these analyses employe models that are amenable to rigorous uncertainty treatment: the forward execution time of model is relatively short (possibly due in part to access to high-performance or high-throughput computing) and/or the number of uncertain model inputs is relatively small. Indeed, to date, numerous uncertainty analysis frameworks have been proposed in the literature, see for example Wu and Liu (2012); Yen et al. (2014); Lu et al. (2014); Wang et al. (2016); most of these approaches employ sophisticated Bayesian analysis techniques, but, as a result of associated computational burden, are limited to models with short execution times and few uncertain model inputs to treat as parameters.

Unfortunately, application of uncertainty analysis techniques to complex, computationally-intensive environmental models (such as groundwater, surface water, and ecosystem models) is difficult because of the large numbers of model inputs that are uncertain, which can easily number in the thousands. This difficulty is compounded by the use of increasingly sophisticated and complex forward models that require increasingly long execution times. The result is large computational requirements, which, for many uncertainty analysis methods, yields an intractable problem. In this situation, computationally-tractable methods are needed (Hill et al., 2016). One class of techniques to evaluate model forecast uncertainty that scales efficiently to high dimensions is linear-based, first-order, second-moment (FOSM) uncertainty analyses,

* Corresponding author.
E-mail address: jwhite@usgs.gov (J.T. White).

http://dx.doi.org/10.1016/j.envsoft.2016.08.017
1364-8152/Published by Elsevier Ltd.

also known as Bayes linear theory (Goldstein and Wooff, 2007). While FOSM-based analyses have been proven successful in the environmental modeling literature (Glasgow et al., 2003; Moore and Doherty, 2005; Gallagher and Doherty, 2007; James et al., 2009; Dausman et al., 2010; Fienen et al., 2010; Lu et al., 2012; White et al., 2014; Leaf et al., 2015; Brakefield et al., 2015), existing software tools for practicing environmental modelers to easily apply these techniques are scant and the understanding required to use these tools can discourage potential users. Existing software tools that perform some form of FOSM-based analyses include PEST++ (Welter et al., 2015), the PREDUNC and PREDVAR software suites (Doherty, 2015), UCODE-2014 utility software (Poeter et al., 2014) and the software OPR-PPR (Tonkin et al., 2007), which was constructed using the building blocks within the joint universal parameter identification and evaluation of reliability application programming interface (JUPITER API) (Banta et al., 2006). Each of these tools provide FOSM as a statically-compiled executable, which is not favorable for exploratory analyses as the executable must be called repeatedly with different inputs and the outputs for each call must be stored/tracked by the user. Additionally, each of these existing tools implement at most two or three calculations, such as parameter uncertainty estimation, forecast uncertainty estimation, or a form of data worth analysis. This places an additional burden on users interested in completing a full suite of FOSM analyses.

Another useful feature of FOSM-based analyses is that they are considered model-independent because these techniques for uncertainty quantification only require the sensitivity of model parameters to model outputs that correspond to observations and forecasts. Therefore, FOSM-based analyses can be applied to any computer model for which these derivatives can be estimated or calculated.

In an effort to remedy the lack of available uncertainty analysis tools that can be easily and efficiently applied to highly-parameterized inverse problems, we have developed a python framework for environmental modeling uncertainty analyses (pyEMU). The pyEMU framework builds on and is compatible with the PEST suite of tools (Doherty, 2015) and has been designed to efficiently implement many forms of FOSM-based analysis in a single framework while also focusing on improving the user experience. pyEMU also implements sophisticated parameter ensemble generation and management capabilities, including the null-space Monte Carlo analysis of Tonkin and Doherty (2009), which uses FOSM theory to pre-condition parameter realization to reduce the computational demand of Monte Carlo analysis.

One of the most attractive uses of FOSM-based analyses is for estimating parameter and forecast uncertainty prior to a computationally-expensive inversion effort, which is possible since linear analyses do not require specific observation values or estimated parameter values. Employing linear analyses prior to inversion allows practitioners to estimate the value of the inversion process to reducing forecast uncertainty, as well as to investigate the sources of uncertainty and the worth of existing and potential new observations. The results of these pre-inversion analyses can be used to guide parameterization design, objective function formulation, and help focus the collection of new data, which are all important elements for any environmental modeling analysis.

pyEMU was created with the goal of increasing the number of environmental modeling analyses that include uncertainty estimates so that model-based resource management decisions can be better informed. This also provides an exploratory computing environment for users to build a better understanding of uncertainty analysis concepts. The software was developed in the context of groundwater modeling, but the techniques are general and can be used with any numerical modeling of environmental systems or others, provided that the model can be driven through text files and that model results can be read from files without manual user intervention.

2. Theory

pyEMU implements several types of linear parameter and predictive uncertainty analyses, including:

- Schur's complement for conditional uncertainty propagation (Koch, 1988; Golub and Van Loan, 1996; Doherty, 2015),
- error variance analysis (Moore and Doherty, 2005; White et al., 2014; Doherty, 2015), including the calculation of parameter identifiability (Doherty and Hunt, 2009; Hill, 2010; Doherty, 2010b), and
- Null Space Monte Carlo (Tonkin and Doherty, 2009).

A brief description of the theory supporting linear-based uncertainty analyses is presented. The interested reader is referred to Doherty (2015) as well as references cited herein for a more complete and rigorous treatment of these concepts.

2.1. Schur's complement

Schur's complement for linear uncertainty analyses can be viewed as a form of Bayes equation under the assumptions of a linear model and multivariate Gaussian distributions to describe stochastic character of parameters, forecasts, and observation noise (Tarantola, 2005; Fienen et al., 2010; Doherty, 2015). Briefly, the posterior parameter covariance matrix, $\overline{\mathbf{\Sigma}}_{\theta}$ estimated with Schur's complement is:

$$\overline{\mathbf{\Sigma}}_{\theta} = \mathbf{\Sigma}_{\theta} - \mathbf{\Sigma}_{\theta}\mathbf{J}^{T}\left[\mathbf{J}\mathbf{\Sigma}_{\theta}\mathbf{J}^{T} + \mathbf{\Sigma}_{\varepsilon}\right]^{-1}\mathbf{J}\mathbf{\Sigma}_{\theta} \tag{1}$$

where $\mathbf{\Sigma}_{\theta}$ is the prior parameter covariance matrix, $\mathbf{\Sigma}_{\varepsilon}$ is the epistemic observation noise covariance matrix, and $\mathbf{J}$ is the Jacobian matrix of partial first derivatives of observations with respect to parameters. This equation highlights the behavior of the inversion process. The first term represents the parameter uncertainty prior to inversion, and the second term encapsulates the inversion process, through the Jacobian matrix and both parameter and observation covariance, as mapping of information from observations to parameters. Depending on which data are collected and their level of certainty, the inversion process should result in a decrease in parameter uncertainty.

Note Eq. (1) expects the user to codify their understanding of the uncertainty in the model parameters – the uncertainty that exists in the parameter before the model is used – in the prior parameter covariance matrix ($\mathbf{\Sigma}_{\theta}$), which is a component of a Bayesian prior description of the parameters. The process of specifying a prior is a foundational part of Bayesian analysis. In this way, the Bayesian form of FOSM analysis is represented by Eq. (1). Note that the Bayesian formulation in pyEMU differs from the forms of FOSM analyses implemented in the OPR-PPR program (Tonkin et al., 2007), UCODE-2014 (Poeter et al., 2014) or the JUPITER API (Banta et al., 2006), which instead implement the inclusion of prior parameter information through the use of prior information equations, rather than through explicit specification of a prior parameter covariance matrix.

The Jacobian matrix is typically calculated using a finite-difference approximation