Journal of Computational Physics 313 (2016) 352–366

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Emulation of dynamic simulators with application to hydrology

David Machac[a,b,*], Peter Reichert[a,b], Carlo Albert[a]

[a] Eawag, Swiss Federal Institute of Aquatic Science and Technology, Department of Systems Analysis, Integrated Assessment and Modelling, 8600 Dübendorf, Switzerland

[b] ETH Zurich, Department of Environmental Systems Science, 8092 Zurich, Switzerland

ARTICLE INFO

Article history:
Received 28 March 2014
Received in revised form 16 November 2015
Accepted 19 February 2016
Available online 23 February 2016

Keywords:
Emulation
Surrogate modeling
Hydrological models

ABSTRACT

Many simulation-intensive tasks in the applied sciences, such as sensitivity analysis, parameter inference or real time control, are hampered by slow simulators. Emulators provide the opportunity of speeding up simulations at the cost of introducing some inaccuracy. An emulator is a fast approximation to a simulator that interpolates between design input–output pairs of the simulator. Increasing the number of design data sets is a computationally demanding way of improving the accuracy of emulation. We investigate the complementary approach of increasing emulation accuracy by including knowledge about the mechanisms of the simulator into the formulation of the emulator. To approximately reproduce the output of dynamic simulators, we consider emulators that are based on a system of linear, ordinary or partial stochastic differential equations with a noise term formulated as a Gaussian process of the parameters to be emulated. This stochastic model is then conditioned to the design data so that it mimics the behavior of the nonlinear simulator as a function of the parameters. The drift terms of the linear model are designed to provide a simplified description of the simulator as a function of its key parameters so that the required corrections by the conditioned Gaussian process noise are as small as possible. The goal of this paper is to compare the gain in accuracy of these emulators by enlarging the design data set and by varying the degree of simplification of the linear model. We apply this framework to a simulator for the shallow water equations in a channel and compare emulation accuracy for emulators based on different spatial discretization levels of the channel and for a standard non-mechanistic emulator. Our results indicate that we have a large gain in accuracy already when using the simplest mechanistic description by a single linear reservoir to formulate the drift term of the linear model. Adding some more reservoirs does not lead to a significant improvement in accuracy. However, the transition to a spatially continuous linear model leads again to a similarly large gain in accuracy as the transition from the non-mechanistic emulator to that based on one reservoir.

* Corresponding author at: Eawag, Swiss Federal Institute of Aquatic Science and Technology, Department of Systems Analysis, Integrated Assessment and Modelling, 8600 Dübendorf, Switzerland.

E-mail address: david.machac@eawag.ch (D. Machac).

http://dx.doi.org/10.1016/j.jcp.2016.02.046
0021-9991/

1. Introduction

In scientific or engineering practice, we often encounter situations where we need to run a computationally complex simulator repeatedly for different inputs (e.g. when doing sensitivity analysis or statistical inference of simulator parameters). This can become infeasible due to excessive computational requirements.

As a workaround to this problem, we can employ a method called *emulation*, which replaces the simulator by a faster surrogate model, called the *emulator*, which is an approximation to the simulator [1]. We may construct the emulator as a *Gaussian process* with empirically specified mean and variance functions and condition this process on n sets of *design data*. Design data are sets of pairs of inputs (or parameter vectors) and corresponding outputs of the simulator, which are required to condition the emulator. After conditioning the stochastic process to the design data, we get an approximation to the simulator outputs for inputs that are not contained in the design data. This basic method of emulation, which does not take into account our knowledge about the simulator, has been developed mainly over the past two decades (e.g. [2–5]).

Several methods have been proposed to emulate simulators with time series output (*dynamic simulators*). The two simplest approaches are to treat time as another input [6], or to emulate the output at all time points jointly as a multivariate output of the emulator (time as output) [7]. These naïve techniques may be adequate for short time series. However, they can either become slow or can run into numerical problems when the time series are long or the time steps are densely spaced. Three approaches have been suggested to cope with these problems within the framework of constructing an emulator by conditioning stochastic processes (see the review by Castelletti [8] for conceptually different approaches). Functional approaches to the emulation of dynamic models are to write the output of the simulation model as a series of local or global basis functions over time and emulate the coefficients of this series as functions of the model inputs [9,10]. If the model output at one time point depends only on the output at the previous time point, we can emulate the transfer function from one time point to the next [11,12]. Finally, the strategy of using a Gaussian process as an innovation term between two successive time points in a dynamic stochastic model was proposed by Liu and West [13].

None of the approaches mentioned above makes use of the knowledge of the simulator mechanisms. This has the advantage that these emulators are very generally applicable. On the other hand, using this knowledge may increase the accuracy of emulation even without extending the design data (which may be computationally demanding). With this motivation, a novel approach, called *mechanism-based emulation*, was suggested that is based on approximately representing the underlying mechanisms of a simulator by a simplified, linear, stochastic model, the noise term of which is formulated as a Gaussian process of the parameters or inputs to be emulated. This stochastic model is then conditioned to the design data, so that the conditioned noise term corrects dynamically for the errors of the linear model and provides interpolation for new parameters or inputs [14,15]. As the Gaussian process, after conditioning, corrects the linear model to mimic the behavior of the nonlinear simulator, this approach works with different underlying linear models. We can expect that a more detailed representation of the mechanisms in the simulator, in particular regarding the dependence on parameters or input, leads to a higher precision of the emulator (given the same design data). On the other hand, the simpler the linear model, the faster the emulator. When constructing such a linear approximation, we have to be aware that this approximation has to be a dynamic, linear model in its state variables for given parameters and input (to facilitate conditioning), but that the coefficients of this model can be nonlinear functions of parameters and input. The better these dependencies are formulated, the less corrections by the Gaussian process are needed. The challenge of constructing the emulator is thus to find the ideal compromise between the number of mechanisms to include in the emulator (this relates to the dimension of the state space and to the dependencies of the coefficients on parameters and input), and emulation speed.

We want to achieve two goals with this paper. First, we further extend the approach by Albert [15] to a continuous-space formulation. Second, we quantify the benefit of incorporating knowledge about the mechanisms underlying the simulator in the approximating linear, stochastic model of the emulator for a specific case study. This case study is based on the diffusive approximation of the *shallow water equations*, a system of two partial differential equations that describe water flow in pipes or open channels. As a simulator, we use a numerical solution of these equations, implemented in Mathematica. This numerical solution exactly copies a single-channel routing component of the Storm Water Management Model (SWMM) by EPA [16] that is used by engineers to predict the outflow from urban drainage basins during and after rain events. Our reason behind using this model is that in our future work, we plan to emulate a complex SWMM model, consisting of a network of channels. We will then also use such a complex model to demonstrate the time gain of emulation. In this case study we will compare emulating this simulator with (i) a mechanistic emulator based on an approximating model in the form of a linear, partial, stochastic differential equation (with a continuous state space; as developed in the first part of the paper), (ii) a mechanistic emulator based on sets of linear, ordinary, stochastic differential equations as developed by Albert [15] (different spatial discretizations leading to discrete state spaces of different dimensions) and (iii) a non-mechanistic emulator. The accuracy of emulation is then investigated as a function of the degree of mechanistic description and of the size of the design data set used for conditioning to quantify the benefit of using our knowledge about the mechanisms of the simulator.