Environmental Modelling & Software 62 (2014) 65–69

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

Short communication

A universal Model-R Coupler to facilitate the use of R functions for model calibration and analysis

Yiping Wu [a, *], Shuguang Liu [b, *], Wende Yan [c]

[a] ASRC Federal, U.S. Geological Survey (USGS), Earth Resources Observation and Science (EROS) Center, Sioux Falls, SD 57198, USA
[b] U.S. Geological Survey (USGS), Earth Resources Observation and Science (EROS) Center, Sioux Falls, SD 57198, USA
[c] National Engineering Laboratory for Applied Forest Ecological Technology in Southern China, Central South University of Forestry and Technology, Changsha 410004, China

ARTICLE INFO

Article history:
Received 20 February 2014
Received in revised form
2 June 2014
Accepted 14 August 2014
Available online

Keywords:
Model calibration
Model environment
OpenMP
PEST
R
R-SWAT

ABSTRACT

Mathematical models are useful in various fields of science and engineering. However, it is a challenge to make a model utilize the open and growing functions (e.g., model inversion) on the R platform due to the requirement of accessing and revising the model's source code. To overcome this barrier, we developed a universal tool that aims to convert a model developed in any computer language to an R function using the template and instruction concept of the Parameter ESTimation program (PEST) and the operational structure of the R-Soil and Water Assessment Tool (R-SWAT). The developed tool (Model-R Coupler) is promising because users of any model can connect an external algorithm (written in R) with their model to implement various model behavior analyses (e.g., parameter optimization, sensitivity and uncertainty analysis, performance evaluation, and visualization) without accessing or modifying the model's source code.

Published by Elsevier Ltd.

Software availability

Name of software: Model-R Coupler
Description: A universal tool developed to allow any model to access various R functions/packages (e.g., model calibration, parameter sensitivity and uncertainty analysis, statistical and graphical evaluations, and instant visualization) without the need for the model's source code.
Developer: Y. Wu, S. Liu, and W. Yan
Source Language: Fortran and R
Availability: http://lca.usgs.gov/lca/nasabioenergy/links.php

1. Introduction

Applying numerical models is a common practice for studying complex dynamic systems in many fields such as biological, physical, chemical, and environmental sciences and engineering, medicine, and economics (Brun et al., 2001; Soetaert and Petzoldt, 2010). To acquire proper simulations or predictive capabilities, a parameter optimization procedure and a model behavior investigation (such as sensitivity/uncertainty analysis) are usually needed for most models including process-based ones (Foglia et al., 2009; Soetaert and Petzoldt, 2010; Zhang et al., 2009).

To conduct these procedures for a specific model, many algorithms are available to the public such as Shuffled Complex Evolution (SCE) (Duan et al., 1992) and pseudo-random search (Price, 1977) algorithms for parameter optimization, and Metropolis-Hastings (MH) (Hastings, 1970) and delayed rejection and adaptive Metropolis (DRAM) (Haario et al., 2006) algorithms for Markov Chain Monte Carlo analysis. However, it is challenging to directly connect an external algorithm (e.g., SCE) or an R package (e.g., Flexible Modeling Environment (FME), which includes several algorithms (Soetaert and Petzoldt, 2010)), with an existing model because it usually requires additional efforts, including accessibility of the model's source code, knowledge of the model's structure and the external algorithm, and additional programming endeavors. Such example is the development of the Soil and Water Assessment Tool (SWAT) (Arnold et al., 1998) Auto-calibration Tool which embeds Fortran-based SCE into the Fortran-based SWAT (Green and van Griensven, 2008; van Griensven, 2006) with substantial

* Corresponding authors.
E-mail addresses: ywu@usgs.gov, rocky.ypwu@gmail.com (Y. Wu), sliu@usgs.gov (S. Liu), csfuywd@hotmail.com (W. Yan).

http://dx.doi.org/10.1016/j.envsoft.2014.08.012
1364-8152/Published by Elsevier Ltd.

programming work (if we check its source code). It is even more challenging when the computer language that is used for an algorithm is different from that used for a model. Such an example is the development of R-SWAT-FME (Wu and Liu, 2012a, 2014). From this perspective, the Model-Independent Parameter Estimation program (PEST) was regarded to be useful because it can implement parameter estimation, sensitivity, and uncertainty analysis for any model without touching the model's source code (Doherty, 2010b). However, the algorithms (e.g., the Marquardt–Levenberg method for parameter estimation) in PEST are fixed and PEST lacks functions for statistical analysis and graphical output. Thus, knowledge of PEST structure and programming efforts are needed to add alternative methods and functions. In spite of these drawbacks in flexibility, statistical analysis, and visualization, PEST is still popular because of its convenience.

Because of the open source feature, a number of functions involving parameter optimization, sensitivity and uncertainty investigations, and statistical and graphical evaluations have been well established using the General Public License (GPL) R software (R Development Core Team, 2009). Such examples are packages like FME, Optmix (Nash and Varadhan, 2011), hydroPSO (Zambrano-Bigiarini and Rojas, 2013), sensitivity (Pujol et al., 2009), hydromad (Andrews et al., 2011), and hydroGOF (Zambrano-Bigiarini, 2013). Considering the popularity and growing capabilities of R, having an existing non-R (i.e., not developed in R) model to access R functions would be attractive, but this involves rewriting the model in R or modifying the model's code to accomplish manipulation of the model and communications between the model and R function (e.g., R-SWAT-FME (Wu and Liu, 2012a, 2014)). Either of the two approaches requires accessing the model's source code, knowing the model's structure, and programming skills of two languages (R and the language the model was written). The development of StellaR can be a step toward addressing this kind of issue because it facilitates translating a Stella model into R (Naimi and Voinov, 2012). However, it would be more promising to make any non-R model able to access potential R functions. With this goal in mind, we developed a universal tool (e.g., the Model-R Coupler) that can allow a model developed in any computer language to be called just like an R function with no need of the source code and thus access potential R packages/functions for a variety of modeling analyses. This paper presents how we developed the Model-R Coupler using the template and instruction concepts of PEST and the operational structure of the R-SWAT (Wu and Liu, 2012a, 2014). We then used a biogeochemical model Erosion Deposition Carbon Model (EDCM) (Liu et al., 2003), a modified version of CENTURY (Parton et al., 1994, 1987), to demonstrate how this tool works.

2. Background

2.1. PEST template and instruction concept

PEST, or PEST++, was developed to assist in data interpretation, model calibration, and predictive analysis (Doherty, 2010a,b; Doherty and Hunt, 2010; Welter et al., 2012). As a nonlinear parameter estimator, PEST can exist independently of any particular model, so it can be used to estimate parameters (i.e., adjusting model parameter in order that the discrepancies between the model simulations and the corresponding measurements are reduced to a minimum) for a wide range of model types—models can be homemade or purchased, small or large, and written in any programming language (Doherty, 2010b). Users do not need to have the source code or know much about the internal workings of a model because PEST uses the template and instruction concept (Doherty, 2010b). The template concept is used to identify which parameters (including which input files) are supposed to be optimized, and the instruction concept recognizes where the output numbers are (including which output files). With the user-specified template and instruction files, PEST can assign a new set of parameter values into their right locations (i.e., writing input files based on a template file), run the model the way it is originally supposed to run (i.e., the command line in the PEST control file), and read the output variables (i.e., read specific numbers as designated in an instruction file). PEST has been successfully used for various fields, especially for hydrological and ecological modeling (Baginska et al., 2003; Finsterle and Zhang, 2011; Gaucherel et al., 2008; Skahill et al., 2009). Details about PEST can be found in the User's Manual (Doherty, 2010b), which is available at http://www.pesthomepage.org/.

2.2. R-SWAT operational structure

To enable the SWAT model to access R functions in the R platform, we converted the Fortran-based SWAT model to an R function (R-SWAT) (Wu and Liu, 2012a). Instead of re-writing the model in R, we used RFortran (Thyer et al., 2011) to accomplish the message transfer between Fortran and R (i.e., parameters from R to Fortran and results from Fortran to R) and provided an outer loop to make the model sleep or run (i.e., the 'model sleep-run loop framework') depending on a signal transfer mechanism between the two computer languages (Wu and Liu, 2012a). The message transfer function provided by RFortran was then replaced by a simple reading/writing method in an updated version of R-SWAT to simplify users' applications (Wu and Liu, 2014). A couple of case studies demonstrated how the R-SWAT can be used to access the comprehensive FME packages for a variety of functionalities including parameter identifiability, optimization, local/global sensitivity, and uncertainty analysis with instant visualization (Wu and Liu, 2012a, 2014). The User's Guide for R-SWAT-FME (Wu and Liu, 2012b) is available at http://pubs.usgs.gov/of/2012/1071/.

3. Model-R Coupler

3.1. Design idea

The PEST template and instruction concept demonstrates that any model can be manipulated 'outside' (without having its source code) to achieve input file re-writing, model running in its original way, and output file reading based on template files, the control file, and instruction files, respectively (see Section 2.1). The R-SWAT operational structure (i.e., 'sleep-run loop framework') indicates that any particular program (not developed in R) can be called like an R function by recasting the program (e.g., SWAT) as a subroutine of the sleep-run loop framework and message transfer (by RFortran or a reading/writing method) between computer languages (see Section 2.2). The PEST does not need the source code of a model, but it cannot access R functions. In contrast, the operational structure of the R-SWAT can help a model utilize the R functions, but it requires the source code of the model in order to embed it into the sleep-run loop framework. Therefore, we proposed combining the advantages of the above two procedures into a single tool, which would allow any model to access R functions (e.g., model calibration and uncertainty analysis with instant visualization) without having the model's source code and without the need for extra programming efforts.

3.2. Development and operation

Driven by the above idea, we borrowed some functions of PEST such as 'input template' and 'output instruction', and then embedded them (denoted as 'PEST concept') into the sleep-run loop framework to formulate a new tool: Model-R Coupler. With the first