Environmental Modelling & Software 68 (2015) 166–174

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

Development of an open-source software package for watershed modeling with the Hydrological Simulation Program in Fortran

D.J. Lampert*, M. Wu [1]

Argonne National Laboratory, 9700 South Cass Avenue, Building 362, Lemont, IL 60439, USA

ARTICLE INFO

Article history:
Received 18 October 2013
Received in revised form
3 February 2015
Accepted 9 February 2015
Available online

Keywords:
Watershed modeling
Python programming
Hydrology
HSPF

ABSTRACT

The Hydrological Simulation Program in Fortran (HSPF) is used extensively for the assessment of water quantity and water quality issues. Herein, the development of an open-source, cross-platform package for building input files, performing simulations, postprocessing and calibrating HSPF models using the Python Programming Language is presented. The flexible nature of Python opens the door to automated preprocessing and calibration routines, visualization, multiprocessing, and larger-scale model development. The software is applied using Python scripts, which provides a flexible mechanism for learning and applying HSPF. An example application of the software was used to build a calibrated HSPF model the Hunting Creek watershed within the Patuxent River Basin, Maryland, USA, which is the example application distributed with the HSPF calibration software package HSPExp. A script of a few hundred lines was used to build a calibrated model comparable to HSPExp in a simulation time of less than two hours.

Software availability

PyHSPF is available on the World Wide Web at:: https://github.com/djlampert/PyHSPF

1. Introduction

Changes in technology, land use, and rapid population growth place an increasing large burden on both the quantity and quality of water in engineered and natural systems. Effective allocation of water resources requires the use of watershed models to simulate the flow of water and the transport of sediment and other associated water quality constituents. The Hydrological Simulation Program in Fortran (HSPF) was developed by the United States Environmental Protection (EPA) in the late 1970s to attempt to fill this need (Singh and Frevert, 2010).

Development of an HSPF model for a watershed requires data on stream flow networks and associated catchment areas, land use, time series of meteorological parameters, and a time series of observed stream flows for calibration. HSPF possesses a flexible, modular design and has been successfully applied to diverse climatic regimes around the world (Singh et al., 1995). HSPF possesses modules for simulating hydrology, temperature, snow, sediment transport, and other quality constituents that can be utilized or ignored as needed. As a result, the model has remained relevant long after its original development (Singh and Frevert, 2010).

HSPF has been jointly maintained by the EPA and the United States Geologic Survey (USGS) up through the current release, Version 12.2 with the EPAs Better Assessment Science Integrating point and Nonpoint Sources (BASINS). The BASINS package provides an integrated Geographic Information Systems (GIS) data system to facilitate the extraction of data, preprocessing of data sets, construction of input files, calibration of model parameters, and visualization of results. Each of these components is essential for the development and application of HSPF models.

While the underlying concept of an integrated preprocessor and modeling platform for BASINS is useful, the BASINS tools are limited. Much of the BASINS graphical user interface (GUI) was written in Visual Basic 6, and thus is not compatible with recent releases of Microsoft Windows based on Visual.NET. Furthermore, the software cannot be used natively on other operating systems (e.g., Unix-like environments). The GUI design of BASINS reduces much of the flexibility inherent to the original modular HSPF design. Attempts to perform simple BASINS tutorials are cumbersome and software crashes are frequent. For these reasons,

* Corresponding author. Tel.: +1 630 252 2533.
E-mail address: djlampert@gmail.com (D.J. Lampert).
[1] Tel.: +1 630 252 6658.

http://dx.doi.org/10.1016/j.envsoft.2015.02.018
1364-8152/

alternative methodologies are needed for preprocessing, simulating, postprocessing, and calibrating HSPF. This paper discusses the development and application of the software "PyHSPF," which is a Python package that can be used for preprocessing, postprocessing, and calibrating HSPF models.

The challenges associated with legacy software such as HSPF are a common problem for environmental modelers. Environmental software packages generally possess unique strengths and weaknesses, and it is often desirable to integrate complementary packages together (Whelan et al., 2014; Voinov and Shugart, 2013). For example, Shrestha et al. (2013) developed an open modeling interface to combine the Soil and Water Assessment Tool (SWAT), a watershed model with sophisticated agricultural runoff and water quality characteristics, and the Storm Water Management Model (SWMM), which possesses a sophisticated representation of river, canal, and sewer hydraulics. Kim et al. (2008) replaced the relatively simplistic groundwater representation in SWAT with the Modular Finite-Difference Flow (MODFLOW) groundwater model to improve SWAT's baseflow runoff representation.

HSPF is the product of many influential studies that remain as relevant today as when they were originally performed. However, the software in its existing form is difficult to apply to new scenarios, and the format and sources of the relevant input data have changed dramatically since the original code was written. The Python programming language contains many useful data analysis tools that are relevant for hydrologic modeling. The PyHSPF package described in this paper combines the hydrology and water quality capabilities of HSPF with data analysis tools in Python to create a flexible framework for the many tasks needed to develop watershed models.

Running an HSPF simulation requires only two input files (though it is possible to use more). The first is the User Control Input (UCI) file, a plain text file that contains instructions for a simulation. The second is the Watershed Data Management (WDM) file, which is an unformatted, direct-access file that contains time series of input data. HSPF Version 12.2 allows multiple WDM files, each of which can store hundreds of time series (Bicknell et al., 2001). Building WDM files requires access to many of the subroutines in the HSPF library. Traditionally this is accomplished with other software such as USGS ANNIE program, or the WDMUtil program that is distributed with BASINS. HSPF uses a message file for runtime error reporting that can be built using the WDIMEX program distributed with the HSPF source code. Herein we describe a new approach to construct the HSPF input files and retrieve simulation results from HSPF simulations.

Calibration of the hydrology process parameters is an important step in developing an HSPF model. To assist in HSPF hydrology calibration, the USGS developed a system called HSPExp (Lumb et al., 1994). HSPExp uses comparisons of observed and simulated flow statistics including total runoff, highest 10% flows, lowest 50% flows, baseflow recession rate, storm flows, and seasonal flows to assist in developing a calibrated HSPF model. Following a simulation, HSPExp provides guidance on what parameters should be increased or decreased and the rationale behind the changes based on the differences in these key statistics. By matching these statistics, the model is expected to demonstrate good consistency with flow–duration curves. HSPExp examines the errors in the simulation results and provides guidance on parameter adjustment until errors are less than the predefined criteria. Herein the performance of the new software tools is compared to HSPExp.

2. Materials and methods

2.1. HSPF library compilation

HSPF Version 12.2 is open source and available online (AQUA TERRA Consultants, 2014). Much of the legacy Fortran 77 code still exists in the current version, including many extraneous utility subroutines designed for pre- and post-processing data. The previous release of HSPF (Version 11) was designed for statically compiling the libraries on Unix-like systems, and the current release contains many artifacts from this approach. The latest release, however, provides no formal make file or detailed instructions to compile the code. The compiled HSPF libraries distributed with BASINS were built with Lahey Fortran, a commercial compiler. Because Fortran 77 does not utilize dynamic memory allocation, the ability to recompile the program is essential (i.e., for development of larger models).

For these reasons, a thorough characterization of the HSPF 11 and 12.2 source codes was performed to determine the essential files needed to build a single HSPF library. A few minor changes were made to the HSPF source code to remove dependencies introduced by extraneous subroutines and to modify some of the file input and output subroutines. No changes were made to the underlying hydrology and water quality process code. The changes to the source code are bundled into an "HSPF13.zip" archive available with the PyHSPF source code in the "misc" directory. The essential Fortran and parameter files from the various LIB3.0 subdirectories were copied into a single directory named "hspf13" along with the source code changes. Collectively the files in this directory can be used to compile 32-bit or 64-bit path-independent shared libraries of essential HSPF subroutines for both for Windows (dynamic-link library, DLL) and Linux (shared object, SO) operating systems with the open source GCC compiler collection (Version 4.6.2) (Stallman, 2003). The resulting libraries from the compilations were used to run the test files distributed with the HSPF Version 12.2 source code. The simulation output was consistent with output files in the source distribution. A batch file (for Microsoft Windows) and a shell script (for Unix-like systems) were developed to facilitate the compilation process. The files require only three calls to the GNU compilers using the files in the hspf13 directory. One call is needed to compile the underlying C code into objects, one to compile the underlying Fortran code into objects, and one to the link the object files into a library. The batch file and shell script are distributed with PyHSPF in the "misc" directory along with a detailed summary of the HSPF Version 12.2 source code and the changes needed to compile the library.

2.2. Python extensions to the HSPF library

The Python Programming Language is a general-purpose, interpreted, high-level language that emphasizes readability (Van Rossum and Drake, 2011). High-level interpreted languages such as Python can be used to integrate together the many tasks involved in building and applying a watershed model. Numeric Python (NumPy) provides a fundamental package for scientific computing with Python including access to multidimensional containers for data. NumPy possesses a Fortran to Python interface generator (F2Py) that can be used to build connections between Python and Fortran (Peterson, 2009).

Python extensions to the HSPF library were developed using a series of Fortran subroutines that act as pointers to the essential HSPF input and output subroutines. The subroutines were placed into a file "hspf_f2py_module.f" with the HSPF source code in the "src/hspf13" directory and include F2PY intent statements that indicate the variables to be used as Python input arguments and the output return expressions. The Python extensions to HSPF were built via an operating system-independent NumPy setup script. The extensions provide access to subroutines that are required to build the WDM files for HSPF. An additional Python script "make_messagefile.py" was written to replace the WDIMEX program to build the HSPF message file. This file is distributed with the PyHSPF source code in the "misc" directory.

2.3. PyHSPF software description

Building an HSPF model requires physical hydrography data on catchment areas and steam reaches, land use category data, time series of meteorological forcing data, and hydrology process parameters for model land segments. The hydrology process parameters are typically calibrated to a particular location, which necessitates time series data of stream flows. All of these data must be gathered together and translated into the HSPF input UCI and WDM files. A series of Python classes were developed to interact with the extensions to HSPF to gather input data, build input and output files, run simulations, calibrate hydrology process parameters, postprocess simulation results, and forecast changes in hydrology under different scenarios. The PyHSPF package is divided into four modules containing a variety of classes and methods for these tasks: the core, preprocessing, calibration, and forecasting modules.

PyHSPF utilizes several third party software packages to gather and process data from publicly-available databases and to postprocess HSPF simulation results. Fig. 1 shows a summary of the PyHSPF dependencies, code development, and the publicly-available datasets for which data collection tools have been developed for HSPF model construction.

2.4. PyHSPF core module

Fig. 2 shows a class usage diagram for the PyHSPF core module. A base "HSPFModel" class was developed to collect the information needed for an HSPF simulation, generate HSPF input and output files and call the main HSPF subroutine after input file creation. HSPF simulates hydrology and water quality using the