Coupling Hydrologic Models with Data Services in an Interoperable Modeling Framework

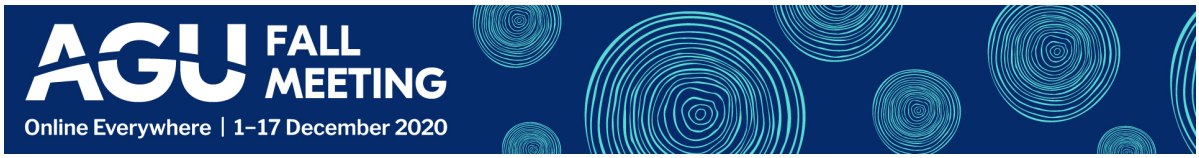
Mark Piper, Rich McDonald, Eric Hutton, Steve Markstrom, Parker Norton, and Greg Tucker

University of Colorado Boulder + U.S. Geological Survey





PRESENTED AT:



PROBLEM

Computational models of flood inundation, precipitation-runoff, and groundwater have traditionally been developed within their individual scientific disciplines. Increasingly, there is a desire and need to couple these models into an integrated system to solve complex problems and aid studies in water resources. For such a system to be viable, individual models must be componentized, becoming self-contained, and exposing a common control interface to the system.

In simpler terms: wouldn't it be neat to take a big, complex model and break it into individual physical process components? We could then conceivably swap out the process components for others (maybe one that employs a novel algorithm or another with a different interpretation of the physics of the process) to study their behavior.

APPROACH

The Community Surface Dynamics Modeling System (CSDMS) (https://csdms.colorado.edu) provides an integrated modeling framework, the CSDMS Workbench (https://csdms.colorado.edu/wiki/Workbench), with software technologies for componentizing, running, and coupling models.

We took the following steps to create and survey a new set of model components.

1. Factored the current development version (v6) of the U.S. Geological Survey (USGS) Precipitation-Runoff Modeling System (PRMS) (https://www.usgs.gov/software/precipitation-runoff-modeling-system-prms) into four independent process components: surface, soil, groundwater, and streamflow. This effort was greatly facilitated by the modern, object-oriented design of PRMS6.
2. Gave each process component, written in Fortran, a Basic Model Interface (BMI) (https://bmi.readthedocs.io), providing a standardized set of functions for querying, modifying, and running the component.
3. Compiled the PRMS components into Python packages with Cython and the Babelizer (https://babelizer.readthedocs.io).
4. Developed a Python package for a service to access gridMET (http://www.climatologylab.org/gridmet.html) climatological data. Like the PRMS components, the data service has a BMI.
5. Coupled the PRMS process components with the Python Modeling Toolkit (PyMT) (pymt.readthedocs.io) and drove the coupled system with climate data from the gridMET data component.

All software developed through this project has been released under open source licenses. The code is available at the GitHub repositories listed below.

PRMS process components (Fortran, with BMI):

- https://github.com/nhm-usgs/bmi-prms6-surface (https://github.com/nhm-usgs/bmi-prms6-surface)
- https://github.com/nhm-usgs/bmi-prms6-soil (https://github.com/nhm-usgs/bmi-prms6-soil)
- https://github.com/nhm-usgs/bmi-prms6-groundwater (https://github.com/nhm-usgs/bmi-prms6-groundwater)
- https://github.com/nhm-usgs/bmi-prms6-streamflow (https://github.com/nhm-usgs/bmi-prms6-streamflow)

gridMET data component (Python, with BMI):

- https://github.com/nhm-usgs/gridmet_bmi (https://github.com/nhm-usgs/gridmet_bmi)

PyMT PRMS components (Python):

- https://github.com/pymt-lab/pymt_prms_surface (https://github.com/pymt-lab/pymt_prms_surface)
- https://github.com/pymt-lab/pymt_prms_soil (https://github.com/pymt-lab/pymt_prms_soil)
- https://github.com/pymt-lab/pymt_prms_groundwater (https://github.com/pymt-lab/pymt_prms_groundwater)
- https://github.com/pymt-lab/pymt_prms_streamflow (https://github.com/pymt-lab/pymt_prms_streamflow)

We factored the USGS Precipitation-Runoff Modeling System (PRMS) into four physical process components, each with a Basic Model Interface.

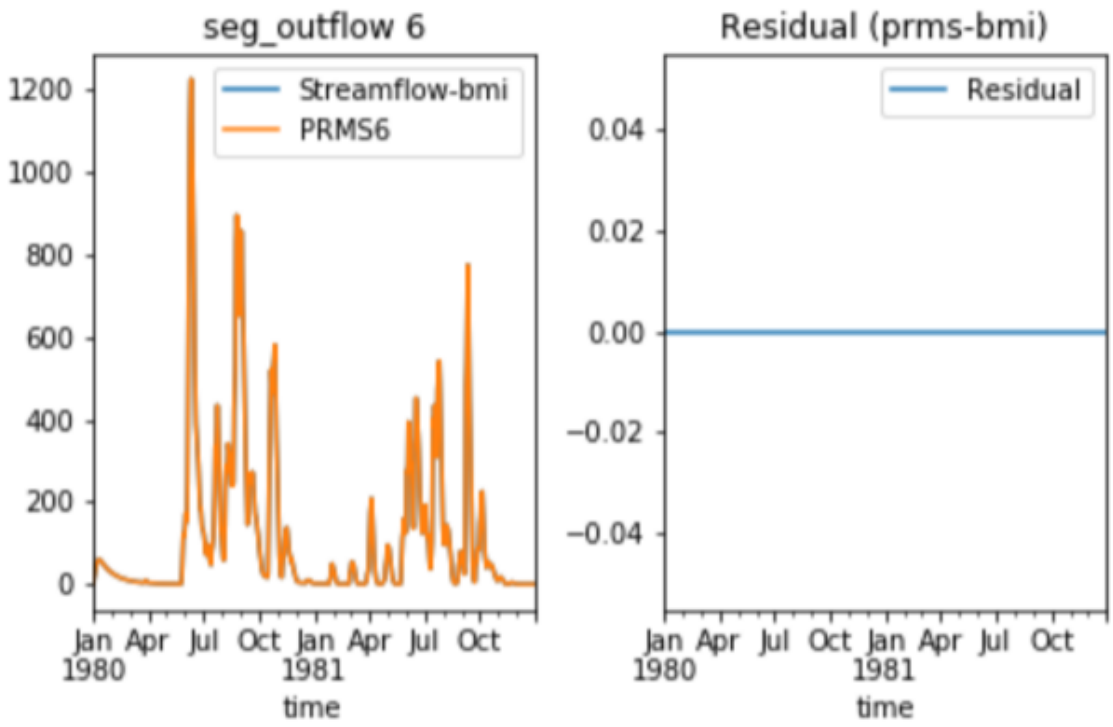
Through Cython, we compiled the components into Python packages.

We developed a package to ingest meteorological data as a service.

We recoupled the PRMS components with the CSDMS Python Modeling Toolkit and drove the system with inputs from the data service.

RESULTS

By recoupling the PRMS components in Python with PyMT, we were able to reproduce the results from running the standalone PRMS model. The figure shows that outflow for the last stream segment in the coupled model system equals that from standalone PRMS.



More detailed examples of coupling the PRMS components in Python with PyMT are demonstrated in the Jupyter Notebooks found in the https://github.com/nhm-usgs/bmi-prms-demo (https://github.com/nhm-usgs/bmi-prms-demo) repository. See the repository README for install and run instructions.

We recently presented this work in a webinar that included a live demonstration of these Jupyter Notebooks.

[VIDEO] https://www.youtube.com/embed/LYPkyKm-Z7Y?rel=0&fs=1&modestbranding=1&rel=0&showinfo=0

DISCUSSION

Just as division of labor allows for increased economic efficiency, component modeling is poised to increase the efficiency of scientific research by allowing the coupling of specialized models into integrated modeling frameworks. Rendering a complex model into independent process components makes it possible to exchange these components for others. Here, we have completed the task of componentizing PRMS.

Next steps include coupling the PRMS process components with other components. Several possibilities exist; for example, a PRMS-MODFLOW coupling where PRMS provides recharge value to MODFLOW, or using PRMS to provide discharge estimates for ungaged tributaries to a hydraulic model such as Delft3D-FM, or couple all three of these models together. Wrapping data-services in BMIs and driving models at runtime with data-streams could eliminate the need for sometimes complicated data formatting and pre-processing.

Additional comments

A Python interface for PRMS makes it easier to use, especially for researchers lacking experience in compiling and linking Fortran code, while PyMT provides a convenient platform for developing and prototyping complex integrated models.

This project was a fruitful collaboration between USGS and University of Colorado researchers, showing that research and operational models written in different languages can be wrapped in Python and coupled in an integrated modeling framework, making them more easily accessible for a new generation of researchers.

Acknowledgements

MP and EH received support from the Department of the Interior, on recommendation from the USGS Community for Data Integration, for this collaboration with RM, SM, and PN.

MP, EH, and GT receive support from the National Science Foundation for their work with CSDMS.

ABSTRACT

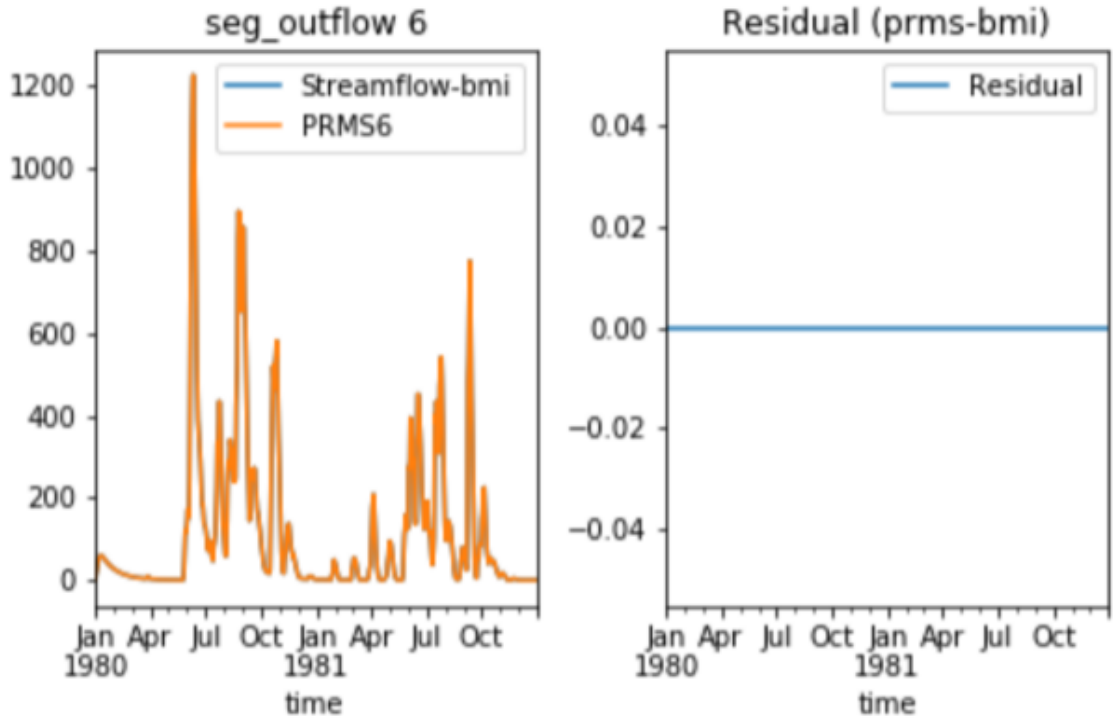
Computational models of flood inundation, precipitation-runoff, and groundwater have traditionally been developed within their individual scientific fields. Increasingly, there is a desire and need to couple these models into an integrated system to solve complex problems and aid studies in water resources; for example, the impact of land-use change or climate variability on surface and subsurface flow in watersheds could be simulated by linking a precipitation-runoff model to a groundwater model.

In this collaborative project, we factored the U.S. Geological Survey (USGS) Precipitation-Runoff Modeling System (PRMS) into four independent process components: surface, soil, groundwater, and streamflow. Each process component, written in Fortran, has a Basic Model Interface (BMI) (https://bmi.readthedocs.io), which gives the model a standardized set of functions allowing it to be queried, modified, and updated in time. When compiled through Cython, the BMI-equipped components become Python packages, and can then be imported into Python with the Python Modeling Toolkit (pymt) (https://pymt.readthedocs.io), which provides a framework and tools for running and coupling models. The addition of a Python interface for PRMS makes it easier to use, especially for researchers lacking experience in compiling and linking Fortran code, and pymt provides an easy collaboration platform for developing and prototyping complex integrated models.

In the next phase of the project, we developed a Python package for a data service to access gridMET climatological data distributed over the web by the University of Idaho. The data service has a BMI, so it can be used directly with pymt for model-data coupling.

Finally, using pymt, we coupled the PRMS process components and drove the coupled system with climate data from the gridMET data component. As a simple test, we were able to reproduce the results from running the standalone PRMS model. (The figure shows that outflow for the last stream segment in the coupled model system equals that from standalone PRMS.)

This project was a fruitful collaboration between USGS and University of Colorado researchers, showing that research and operational models written in different languages can be wrapped in Python and coupled in an integrated modeling framework, making them more easily accessible for a new generation of researchers.



(https://agu.confex.com/data/abstract/agu/fm20/5/0/Paper_682405_abstract_651759_0.png)