The Journal of Open Source Software

Integrated hydrologic model development and postprocessing for GSFLOW using pyGSFLOW

Joshua D. Larsen[1¶], **Ayman Alzraiee**[1], **and Richard G. Niswonger**[2]

1 U.S. Geological Survey, California Water Science Center, Sacramento, CA **2** U.S. Geological Survey, Integrated Modeling and Prediction Division, Water Mission Area, Menlo Park, CA ¶ Corresponding author

DOI: 10.21105/joss.03852

Software

- Review
- Repository
- Archive

Editor: Chris Vernon

Reviewers:

- @thurber
- @mdbartos
- @mdbartos

Submitted: 21 October 2021
Published: 20 April 2022

Overview

pyGSFLOW is a Python package designed to create new GSFLOW integrated hydrologic models, read existing models, edit model input data, run GSFLOW models, process output, and visualize model data.

Introduction

Effective management of limited water resources is challenging because of competing interests and rapid development have resulted in increased demands for water (Garrote, 2017; Lotze-Campen et al., 2006; Mekonnen & Hoekstra, 2016). Understanding interactions among the different components of a hydrologic system, often supported by physics-based models, is important information for water resources planning and management. For example, groundwater models often serve as a tool for exploring the impacts of different water management and conservation scenarios (Davie & Quinn, 2019). The traditional groundwater modeling approach has mainly focused on simulating flow and storage in aquifers that have simplified boundary conditions to represent exchanges with surface water (Chiu et al., 2010; Li & Martin, 2011; Siade et al., 2014). Ignoring spatiotemporal changes in the hydraulically connected surface water and groundwater systems by applying simple and loosely coupled boundary conditions within hydrologic models is no longer adequate to address water management issues (Fatichi et al., 2016). Instead, integrated hydrologic models (IHMs) that couple governing equations for surface-water and groundwater flow are necessary to represent water resources that are impacted by changes in climate and increases in withdrawals and consumption of water.

GSFLOW is an IHM (Markstrom et al., 2008) that simulates surface and subsurface hydrologic processes by integrating the Precipitation Runoff Modeling System (PRMS) (Markstrom et al., 2015) and MODFLOW (Harbaugh, 2005; Niswonger et al., 2011) into a single code that simulates feedbacks between the two systems. Because modelers are moving toward simulating the hydrologic cycle in greater detail, larger datasets from multiple sources are used to parameterize these models. Beyond the scope of example problems, most applied problems require custom workflows and code to process large datasets related to model inputs and outputs. Scripting languages like Python make it easier to process large data sets and provide standard methods that can be used for developing, editing, and properly formatting model input files and for analyzing model output data. These developments have led to major advancements in model reproducibility and improvements in model applicability (Bakker et al., 2016; Gardner et al., 2018; Ng et al., 2018).

Statement of need

GSFLOW model development has previously been a piecemeal approach. Arcpy-GSFLOW scripts (Gardner et al., 2018) or GSFLOW-GRASS packages have been used to process surface-water input data into model files. PRMS-Python (Volk & Turner, 2019) could be used to edit most of the PRMS inputs to GSFLOW. Finally, FloPy (Bakker et al., 2016, 2021) could be used to edit most of the MODFLOW inputs to GSFLOW. This approach is not tightly coupled and still requires manual edits and additional external scripts to edit, run models, and process output data. Because of the complexity of IHMs and the need for model reproducibility, a single integrated scripting package will help standardize and streamline model development and calibration.

pyGSFLOW

pyGSFLOW (Larsen et al., 2021) is a Python package for creating new GSFLOW models, importing existing models, running GSFLOW models, processing model outputs, and visualizing model data. Instead of working directly with formatted model input files, the pyGSFLOW Application Programming Interface (API) allows the user to work with class-based methods to create GSFLOW (Markstrom et al., 2008) and PRMS (Markstrom et al., 2015) input files, MODSIM (Labadie & Larson, 2006) vectorized surface-water operations networks, and MODFLOW (Harbaugh, 2005) model packages and binds them into a single integrated model instance. pyGSFLOW leverages features from FloPy, an existing Python package for the MODFLOW suite of groundwater modeling software (Harbaugh, 2005; Langevin et al., 2017; Niswonger et al., 2011; Panday et al., 2013) and extends the capabilities for IHMs. pyGSFLOW relies on FloPy model and package objects and interfaces with these features to provide FloPy users with familiar code syntax and to ensure the long-term maintainability of the code base.

The pyGSFLOW package was developed for hydrologic modelers and researchers who are developing, calibrating, or applying models as part of their scientific investigation (e.g., prediction scenarios, stream capture) with GSFLOW. The code base currently is being used for the development of several river-basin scale hydrologic models in western basins, including the example shown below highlighting application to the Russian River basin and the Santa Rosa Plain Watershed (figure 1) (Gardner et al., 2018; Woolfenden & Nishikawa, 2014).

The Santa Rosa Plain (SRP) model (Woolfenden & Nishikawa, 2014) is an IHM that was developed as a tool to provide scientific information to water managers about future climate change scenarios. The SRP model applied four global-climate models and simulated relative change in groundwater storage and availability under each scenario. Prior to simulating future change, the model was calibrated to historical groundwater and surface-water conditions. Part of the calibration process involves identifying sensitive and insensitive parameters. Calibration and sensitivity analysis experiments on model parameters can provide insight into reducing model error when predicting results such as simulated streamflow (figure 1). Daily streamflow data from National Water Information System (NWIS) site 11466800 (U.S. Geological Survey, 2022) provided observations for this experiment. Twenty-one iterations of the SRP model were run in series with pyGSFLOW to test the sensitivity of the snarea_curve (snow depletion curve), ssr2gw_rate (gravity reservoir to groundwater reservior routing coeficient), and gwflow_coef (linear groundwater discharge equation coeficient) with a simple "for loop." The ssr2gw_rate was identified as a much more sensitive parameter to model calibration than the snarea_curve and gwflow_coef parameters (figure 1). Insights like these allow researchers to focus their calibration efforts on the most sensitive parameters and fix insensitive parameters, thus reducing the time and complexity of the calibration process. Although actual calibration generally is not done directly with pyGSFLOW, it provides an easy to use interface to update parameters based on grid cell location or parameter zone that can be used for complex operations in conjuction with external calibration software.

The Journal of Open Source Software

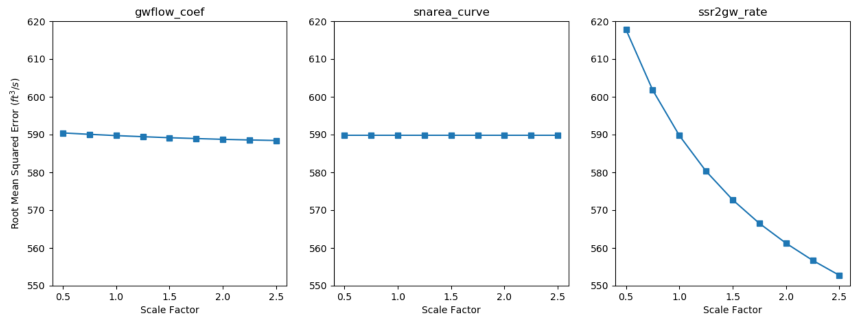

Figure 1: Root mean squared error in streamflow predictions at U.S Geological Survey streamgage 11466800 (U.S. Geological Survey, 2022) for three PRMS parameters (gwflow_coef, snarea_curve, and ssr2gw_rate) during calibration experiments on the Santa Rosa Plain Integrated Hydrologic Model, Santa Rosa, California.

The pyGSFLOW package also includes features to visualize input and output data spatially using Matplotlib (Hunter, 2007) plots and by exporting datasets to shapefile or the visualization toolkit (VTK) format (Schroeder et al., 2006). By providing pyGSFLOW a shapefile or list of hydrologic response unit (HRU) geometries, the code is able to plot arrays and contour arrays of unique parameter values and is fully compatible with the FloPy plotting routines for MODFLOW. PRMS input parameter values can be layered over MODFLOW output and can potentially help identify trends and sensitive parameters controlling those trends in streamflow, recharge, and groundwater levels throughout the model. The ssr2gw_rate parameter, which scales the exchange between the PRMS gravity reservoir and the MODFLOW groundwater reservior in GSFLOW, can then be overlain on top of recharge arrays to inspect the input and output for correlated trends (figure 2). Figure 2 shows that in the western part of the basin, both the ssr2gw_rate and the relative amount of areal recharge is slightly greater than the eastern part of the basin. The simulated recharge data also show the highest volume of recharge occurs along a few short losing stream sections. These insights can help the researcher adjust input parameters for both streamflow and groundwater level calibration.

The Journal of Open Source Software

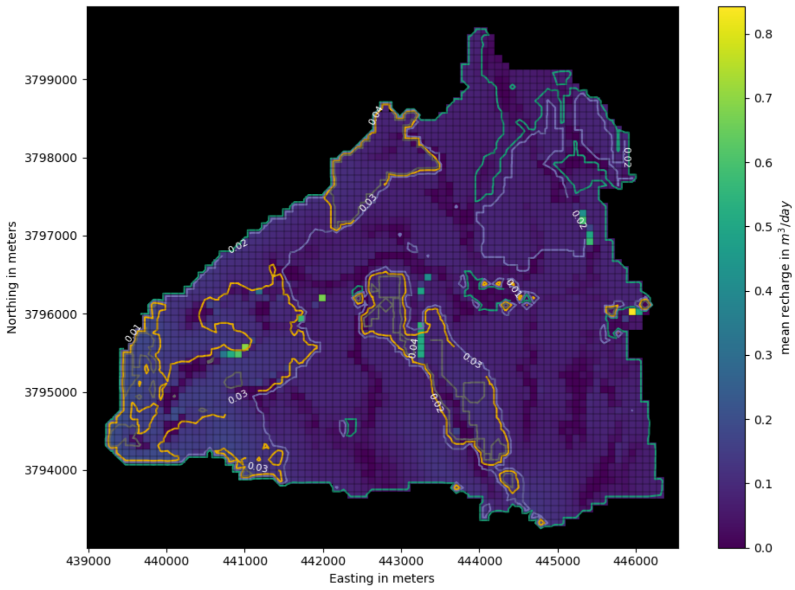

Figure 2: Sagehen Creek GSFLOW model, Truckee, California. Mean recharge for the entire simulation from MODFLOW is overlain with a spatial contour plot of PRMS parameter ssr2gw_rate which is a multiplier that scales the volume of recharge from PRMS to MODFLOW. Black fill indicates inactive model cells.

The online documentation for pyGSFLOW (Larsen et al., 2021) contains API information for all major classes and methods and is updated with each new major release. In addition to the online documentation, sample Jupyter notebooks (Kluyver et al., 2016) are included in the repository to help users become familiar with the code.

Package architecture

The pyGSFLOW package includes the gsflow module and 5 sub-packages (figure 3):

- gsflow: the gsflow module contains the integrated modeling object GsflowModel, which allows the user to build new GSFLOW models and import existing models. This module calls classes and methods from the following 5 sub-packages within pyGSFLOW.
- prms: the prms sub-package contains classes and methods to build new PRMS models, import existing PRMS models, edit model input data, and write PRMS input data to file.
- modflow: the modflow package contains modules and classes that interface with FloPy and allow the user to create new MODFLOW packages, edit existing packages, and write MODFLOW input data to file.
- modsim: the modsim sub-package contains classes that translate MODFLOW model stream and lake networks into vectorized shapefile representations that can be used to define surface-water operation networks in MODSIM.
- output: the output sub-package contains modules that allow the user to define their surface-water model discretization and visualize output data via matplotlib plots.
- utils: includes general use utilities that are integrated into built in functions in the gsflow module, and prms, modflow, and modsim sub-packages.

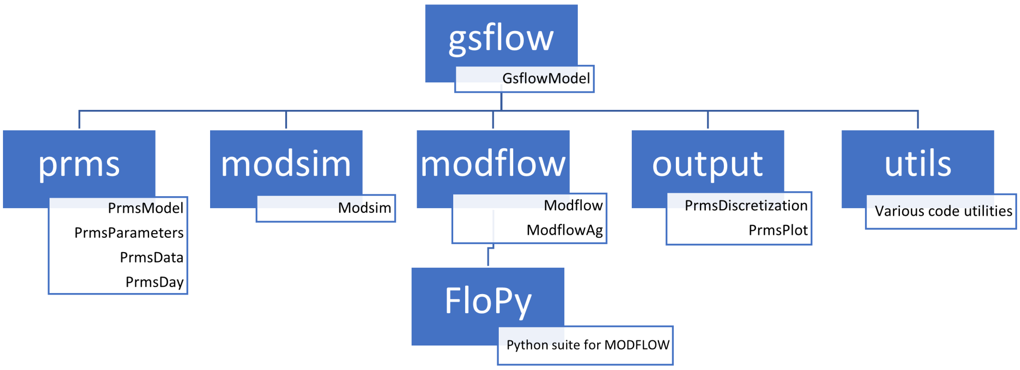

Figure 3: Hierarchical representation of the pyGSFLOW package. Each sub-package lists the model building classes within each package. The GsflowModel class interacts with each of these listed modules and the FloPy package.

Conclusion

GSFLOW IHMs simulate complex processes and interactions between surface-water and groundwater flow systems. Parameterizing these model processes requires large datasets from multiple sources to represent the hydrologic cycle. Previous approaches involved multiple disconnected scripts and packages, some of which rely on proprietary code, which makes reproducibility difficult. pyGSFLOW is a tightly coupled software package, that allows the user to import all parts of their model in one script, which helps to standardize and streamline model development, calibration, and output analysis.

Acknowledgments

The authors welcome additions, suggestions, and assistance from the scientific community, and thank all past contributors for their work. Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Funding

This work was funded by the United States Geological Survey's Water Availability and Use Program.

References

Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J. T., Leaf, A. T., Paulinski, S. R., Larsen, J. D., Towes, M., Morway, E. D., Bellino, J. C., Starn, J. J., & Fienen, M. N. (2021). *FloPy v3.3.3*. U.S. Geological Survey Software Release. https://doi.org/10.5066/F7BK19FH

Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J. T., Starn, J. J., & Fienen, M. N. (2016). Scripting MODFLOW model development using python and flopy. *Groundwater*, *54*, 733–739. https://doi.org/10.1111/gwat.12413

Chiu, Y., Nishikawa, T., & Yeh, W.-G. (2010). Optimal pump and recharge management model for nitrite removal in the warren groundwater basin, california. *Journal of Wa-*

ter Resources Planning and Management, *136*(3). https://doi.org/10.1061/(ASCE)WR.1943-5452.0000034

Davie, T., & Quinn, N. W. (2019). *Fundamentals of hydrology*. Routledge. https://doi.org/10.4324/9780203798942

Fatichi, S., Vivoni, E. R., Ogden, F. L., Ivanov, V. Y., Mirus, B., Gochis, D., Downer, C. W., Camporese, M., Davison, J. H., Ebel, B., Jones, N., Kim, J., Mascaro, G., Niswonger, R., Restrepo, P., Rigon, R., Shen, C., Sulis, M., & Tarboton, D. (2016). *An overview of current applications, challenges, and future trends in distributed process-based models in hydrology*. *537*, 45–60. https://doi.org/10.1016/j.jhydrol.2016.03.026

Gardner, M. A., Morton, C. G., Huntington, J. L., Niswonger, R. G., & Henson, W. R. (2018). Input data processing tools for the integrated hydrologic model GSFLOW. *Environmental Modelling & Software*, *109*, 41–53. https://doi.org/10.1016/j.envsoft.2018.07.020

Garrote, L. (2017). Managing water resources to adapt to climate change: Facing uncertainty and scarcity in a changing context. *Water Resources Management*, *31*, 2951–2963. https://doi.org/10.1007/s11269-017-1714-6

Harbaugh, A. W. (2005). *MODFLOW-2005, the U.S. Geological Survey modular ground-water model – the ground-water flow process*. U.S. Geological Survey. https://doi.org/10.3133/tm6A16

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. (2016). *Jupyter notebooks – a publishing format for reproducible computational workflows* (F. Loizides & B. Schmidt, Eds.; pp. 87–90). IOS Press. https://doi.org/10.3233/978-1-61499-649-1-87

Labadie, J., & Larson, R. (2006). *MODSIM: River basin management decision support system, user manual*. Department of Civil Engineering, Colorado State University, Ft. Collins, CO. http://modsim.engr.colostate.edu/modsim.php

Langevin, C. D., Hughes, J. D., Banta, E. R., Niswonger, R. G., Panday, S., & Provost, A. M. (2017). *Documentation for the MODFLOW 6 groundwater flow model*. U.S Geological Survey. https://doi.org/10.3133/tm6A55

Larsen, J. D., Alzraiee, A., & Niswonger, R. G. (2021). *pyGSFLOW v1.0.0*. U.S. Geological Survey Software Release. https://doi.org/10.5066/P9NPZ5AD

Li, Z., & Martin, P. (2011). *Geohydrology, simulation of regional groundwater flow, and assessment of water management strategies, twentynine palms area, california*. U.S. Geological Survey Scientific Investigations Report. https://doi.org/10.3133/sir20105249

Lotze-Campen, H., Muller, C., Bondeau, A., Stefanie, R., Popp, A., & Lucht, W. (2006). Global food demand, productivity growth, and the scarcity of land and water resources: A spatially explicit mathematical programming approach. *Agricultural Economics*, *39*, 325–338. https://doi.org/10.1111/j.1574-0862.2008.00336.x

Markstrom, S. L., Niswonger, R. G., Regan, R. S., Prudic, D. E., & Barlow, P. M. (2008). *GSFLOW-coupled ground-water and surface-water FLOW model based on the integration of the precipitation-runoff modeling system (PRMS) and the modular ground-water flow model (MODFLOW-2005)*. U.S Geological Survey. https://doi.org/10.13140/2.1.2741.9202

Markstrom, S. L., Regan, R. S., Hay, L. E., Viger, R. J., Webb, R. M. T., Payn, R. A., & H., L. J. (2015). *PRMS-IV, the precipitation-runoff modeling system, version 4*. U.S Geological Survey. https://doi.org/10.3133/tm6B7

Mekonnen, M. M., & Hoekstra, A. Y. (2016). Four billion people facing severe water scarcity. *Science Advances*, *2*(2). https://doi.org/10.1126/sciadv.1500323

Ng, G. H. C., Wickert, A. D., Somers, L. D., Saberi, L., Cronkite-Ratcliff, C., Niswonger, R. G., & McKenzie, J. M. (2018). GSFLOW–GRASS v1.0.0: GIS-enabled hydrologic modeling of coupled groundwater–surface-water systems. *Geoscientific Model Development*, *11*, 4755–4777. https://doi.org/10.5194/gmd-11-4755-2018

Niswonger, R. G., Panday, S., & Ibaraki, M. (2011). *MODFLOW-NWT, a newton formulation for MODFLOW-2005*. U.S Geological Survey. https://doi.org/10.3133/tm6A37

Panday, S., Langevin, C. D., Niswonger, R. G., Ibaraki, M., & Hughes, J. D. (2013). *MODFLOW-USG version 1: An unstructured grid version of MODFLOW for simulating groundwater flow and tightly coupled processes using a control volume finite-difference formulation*. U.S Geological Survey. https://doi.org/10.3133/tm6A45

Schroeder, W., Martin, K., & Lorensen, B. (2006). *The visualization toolkit: An object-oriented approach to 3D graphics. ISBN 978-1-930934-19-1.* Kitware. https://books.google.com/books?id=rx4vPwAACAAJ

Siade, A. J., Nishikawa, T., Rewis, D. L., Martin, P., & Phillips, S. P. (2014). *Groundwater-flow and land-subsidence model of antelope valley, california*. U.S. Geological Survey, Scientific Investigations Report 2014–5166. https://doi.org/10.3133/sir20145166

U.S. Geological Survey. (2022). *USGS water data for the nation: U.S. Geological Survey national water information system database*. https://doi.org/10.5066/F7P55KJN

Volk, J. M., & Turner, M. A. (2019). PRMS-python: A python framework for programmatic PRMS modeling and access to its data structures. *Environmental Modelling & Software*, *114*, 152–165. https://doi.org/10.1016/J.ENVSOFT.2019.01.006

Woolfenden, L. R., & Nishikawa, T. (2014). *Simulation of groundwater and surface-water resources of the santa rosa plain watershed, sonoma county, california*. U.S. Geological Survey Scientific Investigations Report. https://doi.org/10.3133/sir20145052