Modern and Reproducible Groundwater Modeling Workflows with FloPy

Joseph D. Hughes [1,*], Christian D. Langevin [2], Scott R. Paulinski [3], Joshua D. Larsen [4] and David Brakenhoff [5]

[1] *U.S. Geological Survey, Earth Systems Modeling Branch, 927 W Belle Plaine Ave, Chicago, IL, USA*
[2] *U.S. Geological Survey, Earth Systems Modeling Branch, 2280 Woodale Dr, Mounds View, MN, USA*
[3] *U.S. Geological Survey, California Water Science Center, 4165 Spruance Road, Suite 200, San Diego, CA, USA*
[4] *U.S. Geological Survey, California Water Science Center, 6000 J Street, Placer Hall, Sacramento, CA, USA*
[5] *Artesia Water, Korte Weistraat 12, Schoonhoven, Netherlands*

Correspondence*:
Corresponding Author
email@uni.edu

ABSTRACT

For full guidelines regarding your manuscript please refer to Author Guidelines.

As a primary goal, the abstract should render the general significance and conceptual advance of the work clearly accessible to a broad readership. References should not be cited in the abstract. Leave the Abstract empty if your article does not require one, please see Summary Table for details according to article type.

Keywords: MODFLOW, FloPy, groundwater model, python, keyword, keyword, keyword, keyword

1 INTRODUCTION

FloPy is a popular Python package for constructing, running, and post processing MODFLOW-based groundwater models. It is open source and developed with input from a growing community of modelers.

It has been recommended as one way to facilitate repeatable research and sharing of ideas (Fienen and Bakker, 2016)

We have FloPy to be particularly useful for teaching. Annotated Jupyter notebooks, comparison with analytical solutions, ...

We rely on FloPy for MODFLOW development. We write tests that rely on FloPy to construct and run models, and then read output. We then verify that the output is as expected, by using the results from an analytical solution, results from another model, or results that have been confirmed to be correct.

We use FloPy to load models

FloPy is used to pioneer new methods and analysis tools, such as deep learning approaches for improving groundwater model calibration (Sun, 2018; Zhou and Tartakovsky, 2021), regionalization of residence times using metamodeling (Starn and Belitz, 2018), iterative ensemble approaches for calibration and uncertainty quantification (White, 2018), and exploration of alternative parameterization schemes for risk analysis (Knowling et al., 2019). There are numerous examples of constructing MODFLOW models to solve applied groundwater problems (Befus et al., 2017; van Engelen et al., 2018; Ebeling et al., 2019; Zipper et al., 2019; Befus et al., 2020). Used in GIS-based tools, such as FREEWAT (Rossetto et al., 2018) and other cyberinfrastructures (Essawy et al., 2018) to export models into MODFLOW datasets. FloPy can also be used as the "glue" to help couple MODFLOW to other hydrological models (Burek et al., 2020) or even to agent-based models designed to quantify the effects of decision makers on environmental behavior (Jaxa-Rozen et al., 2019).

Bakker et al. (2016) describe the general approach for working with models within the python environment and emphasize the reproducible nature of developing models through scripting. FloPy has continued to advance since it was first described by Bakker et al. (2016). The purpose of this paper is to highlight some of these important advances, provide examples that demonstrate these new capabilities, and reinforce the advantages of the modern scripting workflow for developing reproducible groundwater models that can be easily updated as new data become available. The important advances described here can be summarized as

- expanded support for MODFLOW-2005, MODFLOW-NWT, MODFLOW-LGR, MODFLOW-USG, SEAWAT, MT3D, MT3D-USGS
- rapid and robust support for all MODFLOW 6 models, packages, and options,
- generalized support for structured and unstructured model grids,
- implementation of new geoprocessing capabilities to rapidly populate models with data,
- export capabilities for writing model data to a variety of output formats,
- plotting capabilities for map and cross-section views of model data, and
- simplified access to model results.

2 FLOPY SUPPORT FOR MODFLOW 6

The most recent version of MODFLOW (MODFLOW 6) is an object-oriented program and framework developed to provide a platform for supporting multiple models and multiple types of models within the same simulation (Langevin et al., 2017; Hughes et al., 2017; Morway et al., 2021). These models can be independent of one another with no interaction, they can exchange information, or they can be tightly coupled at the matrix level by adding them to the same numerical solution. Transfer of information between models is isolated to exchange objects, which allow models to be developed and used independently. Within this new framework, a regional-scale groundwater model may be coupled with multiple local-scale groundwater models.

MODFLOW 6 currently includes the Groundwater Flow (GWF) Model and the Groundwater Transport (GWT) Model each with packages to represent surface water processes, groundwater extraction, external boundaries, mass sources and sinks, and mass sorption and reactions. GWF and GWT models can be developed using regular model grids consisting of layers, rows, and columns or they can be developed using more general unstructured grids using many of the concepts and numerical approaches available in MODFLOW-USG (Panday et al., 2013). MODFLOW 6 also includes advanced formulations to simulate three-dimensional anisotropy and dispersion (Provost et al., 2017), coupled variable-density groundwater flow and transport (Langevin et al., 2020), and a water mover package to represent natural and managed hydrologic connections (Morway et al., 2021).

Development and testing of the MODFLOW 6 program relies heavily on tight integration with FloPy. A key component of this tight integration is the capability to quickly support new MODFLOW 6 models and packages with FloPy. Unlike the FloPy support for previous MODFLOW versions (for example, MODFLOW-2005, MODFLOW-NWT, MODFLOW-USG, and SEAWAT), the FloPy python classes for MODFLOW 6 are dynamically generated from simple text files that describe the input file structure. This allows MODFLOW 6 developers to write tests for new models, packages, and functionality as they are developed. All MODFLOW 6 model input files are described using "definition files." These definition files are used to generate the user input and output guide. These same definition files are also used to generate FloPy classes, with argument docstrings corresponding to input variable descriptions in the input and output guide.

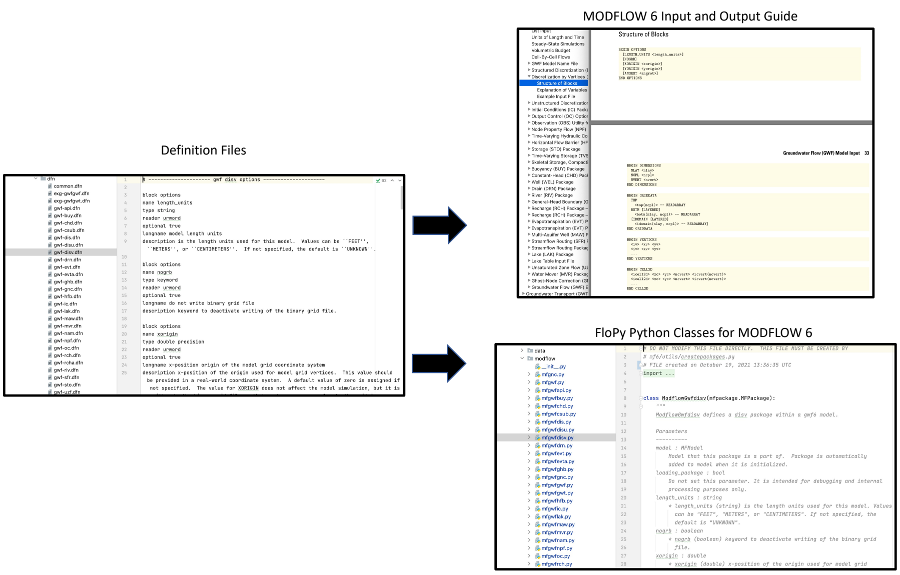

Figure 1. Relation between MODFLOW 6 definition files and the MODFLOW 6 input and output guide and the FloPy Python classes for MODFLOW 6.

3 COMMON MODELING TASKS

3.1 Generating Grids

Support for a variety of different structured and unstructured grid types has been a recent focus of MODFLOW development (Panday et al., 2013; Langevin et al., 2017; Provost et al., 2017). FloPy has been updated to support processing of different types of grids, such as those shown in figure 2 through implementation of several different grid classes. The regular MODFLOW grid consisting of layers, rows, and columns, continues to be a popular type of MODFLOW grid. Regular MODFLOW grids can have constant row and column spacings, as shown in Figure 2A, or they can have variable row and column spacings to focus resolution around an area of interest, as shown in Figure 2B. FloPy internally represents this type of grid as a StructuredGrid object.

Nested grid (Mehl and Hill, 2013) quadtree grid refinement (Lien et al., 2014); then cite triangle (Shewchuk, 1996) and then cite voronoi grid (Virtanen et al., 2020) cite any groundwater papers? algomesh?).

FloPy gridding allows for innovation; mention Central Sands and the ability to simulate local-scale detail and regional-scale influence in the same simulation?

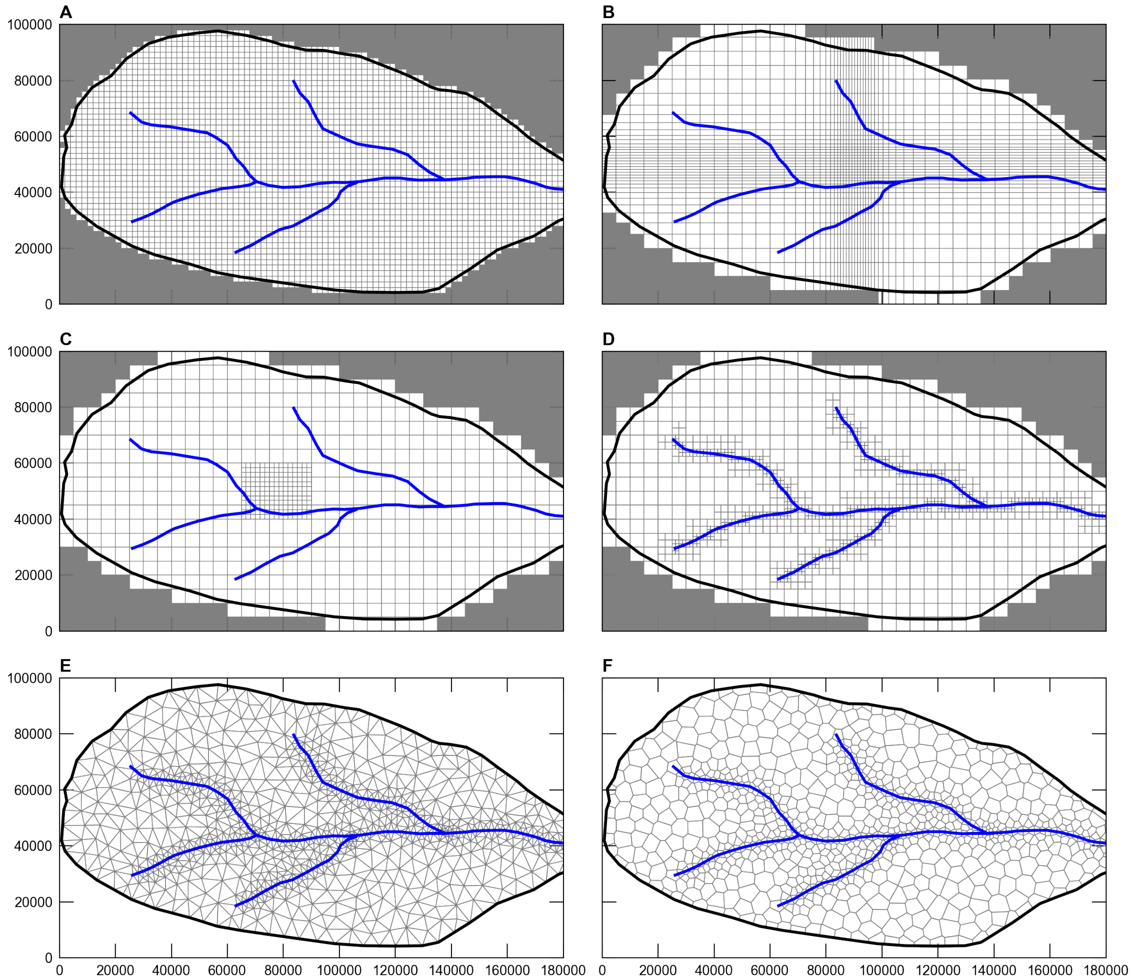

Figure 2. Examples of grids that can be generated and processed using FloPy, including (A) a regular structured MODFLOW grid, (B) a structured MODFLOW grid with irregular spacing, (C) a regular MODFLOW child grid nested within a regular MODFLOW parent grid, (D) a quadtree grid, (D) a triangular grid, and (E) a voronoi grid

3.2 Geospatial Processing

Intersections, raster resampling, ...

3.3 Plotting

3.4 Exporting Grid Data to Other Formats

shapefiles (all grids), VTK (all grids) and NetCDF (structured grids)

4 EXAMPLE

Background of the McDonald Valley

5 DISCUSSION AND CONCLUSIONS

FloPy is a popular python package for building, running, and post processing groundwater models. It is open source and developed with input from a growing community of modelers.

Key findings

- FloPy fully supports creation and loading of all MODFLOW 6 models and packages. FloPy classes can be built and updated automatically using MODFLOW 6 definition files, which describe input format. FloPy also supports MODFLOW-2005, MODFLOW-NWT, MODFLOW-USG, MT3D, and MT3D-USGS.
- FloPy contains a low-level Grid class, which can be used to represent regular MODFLOW grids consisting of layers, rows, and columns, or unstructured grids consisting of vertices and incidence lists. The Grid class is used systemically throughout FloPy for geospatial operations, plotting, and exporting model information to supported formats.
- Geospatial intersections of points, lines, and polygons with model grids and raster resampling onto model grids are common steps in model construction. FloPy fully supports these geospatial operations through its grid intersection and raster resampling routines.
- Map and cross section plotting
- Export to shapefiles, VTK, and NetCDF

ACKNOWLEDGMENTS

This is a short text to acknowledge the contributions of specific colleagues, institutions, or agencies that aided the efforts of the authors.

SUPPLEMENTAL DATA

Supplementary Material should be uploaded separately on submission, if there are Supplementary Figures, please include the caption in the same file as the figure. LaTeX Supplementary Material templates can be found in the Frontiers LaTeX folder.

DATA AVAILABILITY STATEMENT

The datasets [GENERATED/ANALYZED] for this study can be found in the [NAME OF REPOSITORY] [LINK].

REFERENCES

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

Befus, K. M., Barnard, P. L., Hoover, D. J., Finzi Hart, J. A., and Voss, C. I. (2020). Increasing threat of coastal groundwater hazards from sea-level rise in california. *Nature Climate Change* 10, 946–952

Befus, K. M., Kroeger, K. D., Smith, C. G., and Swarzenski, P. W. (2017). The magnitude and origin of groundwater discharge to eastern u.s. and gulf of mexico coastal waters. *Geophysical Research Letters* 44, 10,396–10,406. doi:https://doi.org/10.1002/2017GL075238

Burek, P., Satoh, Y., Kahil, T., Tang, T., Greve, P., Smilovic, M., et al. (2020). Development of the community water model (cwatm v1.04) – a high-resolution hydrological model for global and regional assessment of integrated water resources management. *Geoscientific Model Development* 13, 3267–3298. doi:10.5194/gmd-13-3267-2020

Ebeling, P., Handel, F., and Walther, M. (2019). Potential of mixed hydraulic barriers to remediate seawater intrusion. *Science of The Total Environment* 693, 133478. doi:https://doi.org/10.1016/j.scitotenv.2019.07.284

Essawy, B. T., Goodall, J. L., Zell, W., Voce, D., Morsy, M. M., Sadler, J., et al. (2018). Integrating scientific cyberinfrastructures to improve reproducibility in computational hydrology: Example for hydroshare and geotrust. *Environmental Modelling & Software* 105, 217–229. doi:https://doi.org/10.1016/j.envsoft.2018.03.025

Fienen, M. N. and Bakker, M. (2016). Hess opinions: Repeatable research: what hydrologists can learn from the duke cancer research scandal. *Hydrology and Earth System Sciences* 20, 3739–3743. doi:10.5194/hess-20-3739-2016

Hughes, J. D., Langevin, C. D., and Banta, E. R. (2017). *Documentation for the MODFLOW 6 framework*. U.S. Geological Survey Techniques and Methods, book 6, chap. A57, 36 p. doi:10.3133/tm6A57

Jaxa-Rozen, M., Kwakkel, J. H., and Bloemendal, M. (2019). A coupled simulation architecture for agent-based/geohydrological modelling with netlogo and modflow. *Environmental Modelling & Software* 115, 19–37. doi:https://doi.org/10.1016/j.envsoft.2019.01.020

Knowling, M. J., White, J. T., and Moore, C. R. (2019). Role of model parameterization in risk-based decision support: An empirical exploration. *Advances in Water Resources* 128, 59–73. doi:https://doi.org/10.1016/j.advwatres.2019.04.010

Langevin, C. D., Hughes, J. D., Provost, A. M., Banta, E. R., Niswonger, R. G., and Panday, S. (2017). *Documentation for the MODFLOW 6 Groundwater Flow (GWF) Model*. U.S. Geological Survey Techniques and Methods, book 6, chap. A55, 197 p. doi:10.3133/tm6A55

Langevin, C. D., Panday, S., and Provost, A. M. (2020). Hydraulic-head formulation for density-dependent flow and transport. *Groundwater* 58, 349–362

Lien, J.-M., Liu, G., and Langevin, C. D. (2014). *GRIDGEN Version 1.0: A computer program for generating unstructured finite-volume grids*. U.S. Geological Survey Open-File Report 2014–1109, 26 p. doi:10.3133/ofr20141109

Mehl, S. W. and Hill, M. C. (2013). *MODFLOW-LGR—Documentation of ghost node local grid refinement (LGR2) for multiple areas and the boundary flow and head (BFH2) package*. U.S. Geological Survey Techniques and Methods, book 6, chap. A44, 43 p.

Morway, E. D., Langevin, C. D., and Hughes, J. D. (2021). Use of the MODFLOW 6 water mover package to represent natural and managed hydrologic connections. *Groundwater* doi:10.1111/gwat.13117

Panday, S., Langevin, C. D., Niswonger, R. G., Ibaraki, M., and 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 Techniques and Methods, book 6, chap. A45, 66 p.

Provost, A. M., Langevin, C. D., and Hughes, J. D. (2017). *Documentation for the "XT3D" Option in the Node Property Flow (NPF) Package of MODFLOW 6*. U.S. Geological Survey Techniques and Methods, book 6, chap. A56, 46 p. doi:10.3133/tm6A56

Rossetto, R., De Filippis, G., Borsi, I., Foglia, L., Cannata, M., Criollo, R., et al. (2018). Integrating free and open source tools and distributed modelling codes in gis environment for data-based groundwater management. *Environmental Modelling & Software* 107, 210–230. doi:https://doi.org/10.1016/j.envsoft.2018.06.007

Shewchuk, J. R. (1996). Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In *Applied Computational Geormetry, Towards Geometric Engineering, FCRC'96 Workshop, WACG'96, Philadelphia, PA, USA, May 27-28, 1996, Selected Papers*. 203–222. doi:10.1007/BFb0014497

Starn, J. J. and Belitz, K. (2018). Regionalization of groundwater residence time using metamodeling. *Water Resources Research* 54, 6357–6373. doi:https://doi.org/10.1029/2017WR021531

Sun, A. Y. (2018). Discovering state-parameter mappings in subsurface models using generative adversarial networks. *Geophysical Research Letters* 45, 11,137–11,146. doi:https://doi.org/10.1029/2018GL080404

van Engelen, J., Oude Essink, G. H., Kooi, H., and Bierkens, M. F. (2018). On the origins of hypersaline groundwater in the nile delta aquifer. *Journal of Hydrology* 560, 301–317. doi:https://doi.org/10.1016/j.jhydrol.2018.03.029

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17, 261–272. doi:10.1038/s41592-019-0686-2

White, J. T. (2018). A model-independent iterative ensemble smoother for efficient history-matching and uncertainty quantification in very high dimensions. *Environmental Modelling & Software* 109, 191–201. doi:https://doi.org/10.1016/j.envsoft.2018.06.009

Zhou, Z. and Tartakovsky, D. M. (2021). Markov chain monte carlo with neural network surrogates: application to contaminant source identification. *Stochastic Environmental Research and Risk Assessment* 35, 639–651

Zipper, S. C., Gleeson, T., Kerr, B., Howard, J. K., Rohde, M. M., Carah, J., et al. (2019). Rapid and accurate estimates of streamflow depletion caused by groundwater pumping using analytical depletion functions. *Water Resources Research* 55, 5807–5829. doi:https://doi.org/10.1029/2018WR024403

FIGURE CAPTIONS

Figure 3. Enter the caption for your figure here. Repeat as necessary for each of your figures