The Journal of Open Source Education

Self-Guided Decision Support Groundwater Modelling with Python

Rui T. Hugman [1], **Jeremy T. White** [1], **Mike Fienen** [2], **Brioch Hemmings** [3], **and Katie Markovitch** [1]

1 INTERA Geosciences, Perth, Western Australia, Australia **2** U.S. Geological Survey, Upper Midwest Water Science Center, Madison, WI USA **3** Wairakei Research Centre, GNS Science, Taupō, New Zealand

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Submitted: 23 October 2023
Published: unpublished

Summary

The GMDSI tutorial notebooks repository provides learners with a comprehensive set of tutorials for self-guided training on decision-support groundwater modelling using Python-based tools. Although targeted at groundwater modelling, they are based around model-agnostic tools and readily transferable to other environmental modelling workflows. The tutorials are divided into three parts. The first covers fundamental theoretical concepts. These are intended as background reading for reference on an as-needed basis. Tutorials in the second part introduce learners to some of the core concepts parameter estimation in a groundwater modelling context, as well as providing a gentle introduction to the `PEST`, `PEST++` and `pyEMU` software. Lastly, the third part demonstrates how to implement highly-parameterized applied decision-support modelling workflows. Their aim is to provide examples of both "how to use" the software as well as "how to think" about using the software. A key advantage to using notebooks in this context is that the workflows described run the same code as practitioners would run on a large-scale real-world application. Using a small synthetic model facilitates rapid progression through the workflow.

Statement of Need

Effective environmental management necessitates transparent acknowledgment of uncertainties in critical decision-making predictions, coupled with efforts to mitigate these uncertainties, especially when significant risks accompany management outcomes. The significance of uncertainty quantification (UQ) and parameter estimation (PE) in environmental modeling for decision support is widely acknowledged. UQ provides estimates of outcome uncertainty, while PE reduces this uncertainty through assimilating data.

Implementing highly-parameterized UQ and PE in real-world modeling can be challenging due to both theoretical complexity and practical logistics. Limited project time and funding also often hinder their application. Open-source software such as `PEST` (Doherty, 2015) and `PEST++` (Jeremy T. White, Hunt, et al., 2020) provide tools for undertaking UQ and PE analyses. However, the steep learning curve associated with their use and the lack of user-friendly training materials have been a barrier to uptake.

There is a growing demand within the environmental modelling community for transparent, reproducible, and accountable modeling processes, driven by the need for increased credibility and rigor in computational science and environmental simulation (Fienen & Bakker, 2016; J. White et al., n.d.). While some script-based tools enhance the reproducibility of forward model construction (Bakker et al., 2016), they often overlook

UQ and PE analyses. In decision-support scenarios, these analyses are equally vital for robust model deployment as the forward model itself.

The uptake of Python for environmental modeling has increased in recent years, due to its open-source nature, user-friendly syntax, and extensive scientific libraries. Python-based tools have been developed to facilitate UQ and PE analyses, such as `pyEMU` (Jeremy T. White et al., 2016; Jeremy T. White et al., 2021). `pyEMU` is a Python package that provides a framework for implementing UQ and PE analyses with PEST and PEST++. It offers a range of capabilities, including parameter estimation, uncertainty analysis, and management optimization. Although initially designed for groundwater modeling, `pyEMU`'s methodologies are versatile and can be applied to diverse numerical environmental models, as long as they can be manipulated using text files and generate outputs that can be automatically extracted without manual interference.

The tutorial notebooks discussed herein provide a comprehensive, self-guided, and open-source resource for learning decision-support modeling workflows with Python. They are designed to be accessible to a broad audience, including students, researchers, and practitioners who aim to undertake applied environmental decision-support modelling.

Story of the Project

The Groundwater Modelling Decision Support Initiative (GMDSI) is an industry-backed and industry-aligned initiative. Established in mid-2019, its primary goal is to enhance the role of groundwater modeling in groundwater management, regulatory processes, and decision-making. At the core of GMDSI's mission lies the numerical simulation of groundwater movement and processes. Often, data related to groundwater are limited, leading to uncertainties in simulator predictions. However, despite this uncertainty, decisions must be made, and associated risks must be assessed. Modelling plays a central role in the evaluation of these risks.

GMDSI is dedicated to promoting, facilitating, and providing support for the improved utilization of modeling in decision support processes. Its activities endeavor to elevate the role of groundwater modeling in decision-making processes, recognizing the importance of model partner software for UQ and PE, and offering a range of activities aimed at industry engagement, education, practical examples, research, and software development.

A majority of groundwater modelers typically rely on Graphical User Interfaces (GUIs) for their modeling needs. However, each GUI has its unique characteristics and varying degrees of compatibility with external software like `PEST` and `PEST++`. Creating educational materials for these GUIs would necessitate tailoring content to each GUI's specific features, obtaining cooperation from the GUI developers themselves and potentially lagging behind the latest developments.

Decision-support modeling often demands capabilities that surpass what current GUIs can offer. For example, many of GMDSI's worked examples rely on custom-designed utilities or the integration of different software components. Currently, a significant portion of users may not have the expertise to independently implement such advanced approaches. Furthermore, the manual preparation of input files for implementing these complex workflows can be time-consuming. Programmatic workflows, such as those facilitated by `pyEMU`, offer advantages by reducing the time and user input required for setup and execution. This approach is somewhat analogous to the role played by a GUI but offers added flexibility, allowing users to customize and design their own functions and utilities as needed. However, it comes with the drawback of increased potential for user-introduced errors.

Over time, more modelers are turning to Python packages like `FloPy` and `pyEMU` for model and `PEST++` setup. Unfortunately, the adoption of this approach is hindered by a steep

The Journal of Open Source Education

learning curve primarily due to the scarcity of user-friendly training materials. The GMDSI tutorial notebooks aim to address this gap by providing a comprehensive, self-guided, and open-source resource for learning decision-support modeling workflows with Python.

The roots of the materials making up the tutorial notebooks were from a traditional, week-long classroom course curriculum developed for internal training at the USGS by a subset of the authors of this paper. For this course, the instructors leveraged the power of jupyter notebooks as a mechanism to teach both the fundamental background and application of inverse theory. High-level mathematical libraries in python (and other high-level languages with easy plotting utilities such as MATLAB and R) provide an opportunity for students to explore linear algebra and statistical modeling principles that underlie the PE and UQ techniques implemented in `PEST` and `PEST++`. Furthermore, the combination of text, code, and graphics provide an interactive platform for mixing theory and applications and, potentially, providing a template for application on real-world applications. The native support for python makes the connection between worked examples and notebooks seamless and has connections with other worked examples [Jeremy T. White, Foster, et al. (2020); J. T. White et al. (2020); Fienen et al. (2022); https://github.com/doi-usgs/neversink_workflow]

After three iterations of teaching the in-person class, the instructors concluded that the materials and approach were valuable, but came to question the level of retention by students in a 40-hour intensive setting. It is well-documented that without repetition and rapid adoption of new techniques, they can fade quickly from memory (Glaveski, 2019). As a result, the authors, with support from the GMDSI, endeavored to build on the positive aspects of using jupyter notebooks and explore alternative teaching environments instead of week-long classes. The first major change was to add sufficient narration and explanation to the notebooks to improve possibilities for self-study. The initial design through in-person instruction was to have the notebooks serve as illustrations to assist in a narrative discussion, so bolstering of the explanatory text was necessary to help them stand alone. The next change was to refactor the organization from a strictly linear progression to the current three-part organization discussed below. This led to a hybrid model of self-study punctuated by discussion and background lectures online.

Resources

A webinar hosted by GMDSI introducing the tutorial notebooks can be viewed here. During the webinar the authors provided an overview of the notebooks, as well as a demonstration of how to use them and introduced an online self-guided course.

The GMDSI web-page also hosts an extensive range of resources and educational material on decision support modelling. These include numerous instructional video lectures, webinar recordings, non-programmatic workflow tutorials, as well as worked example reports describing real-world applications.

Software from the `PEST` suite can be downloaded from John Doherty's web page here. The user manual contains much useful information. The PEST Book is also a great resource for learning about the theory underpinning use of the software.

Software from the `PEST++` suite can be accessed from GitHub repository. The user manual contains much useful information, as well as theoretical background to the software. Further theoretical background is available in (Jeremy T. White, Hunt, et al., 2020).

`pyEMU` can be accessed from the Git-Hub repository. The repository contains several example jupyter notebooks. The tutorial notebooks discussed herein provide a more exhaustive and structured learning experience.

The Journal of Open Source Education

Contents and Instructional Design

The tutorial notebooks are structured into three main parts:

Part 0: Introductory Background

Part 0 serves as the foundation, providing essential background material. Learners are encouraged to reference notebooks in Part 0 to polish their understanding of concepts they encounter in Parts 1 and 2. Part 0 is not intended to be a comprehensive resource for all background material, but rather to establish a solid understanding of the basics. The explanations of mathematical concepts are intended to be accessible through visualization and descriptions related to everyday concepts and modelling concepts.

Each notebook in Part 0 is standalone and covers a unique topic. These include: - Introduction to a synthetic model known as the "Freyberg" model. This model is used as a consistent example throughout the tutorial exercises, allowing learners to apply concepts in a practical context. - An introduction to the `pyemu` Python package that is used to complement and interface with PEST/PEST++. - Explanation of fundamental mathematical concepts that are relevant and will be encountered throughout the tutorial notebooks.

Part 1: Introduction to PEST and the Gauss-Levenberg Marquardt Approach

Part 1 focuses on the Gauss-Levenberg Marquardt (GLM) approach to parameter estimation and associated uncertainty analysis in a groundwater modelling context. This was the foundation of the `PEST` software for multiple decades and the theory continues to resonate through newer techniques.

Part 1 is designed to be accessible without strict sequential dependencies. Learners have the flexibility to explore its contents in any order that suits their preferences or needs. These include: - Introduction to concepts such as non-uniqueness, identifiability, and equifinality. - Introduction to the PEST control file and the PEST/PEST++ interface. - Exploring the challenges of parameterization schemes on predictive ability, as well as how to mitigate them. - Introducing first-order second-moment (FOSM) and prior Monte Carlo uncertainty analysis approaches.

While Part 1 notebooks can be largely run in any order, the curriculum was initially designed to start with simple parameterization of a model and to build complexity intentionally throughout the progression of the sequence. The ramifications of simplification and the value of adding complexity are evaluated in the context of the performance of the model in forecasts made outside the parameter estimation conditions. This progression motivates the value of a highly-parameterized approach which is the starting point for many new projects, as explored in Part 2.

Part 2: Python-based Decision-Support Modelling Workflows

Part 2 expands on the foundational knowledge gained in Part 1 and delves into advanced topics related to ensemble-based parameter estimation, uncertainty analysis and optimization methods. These advanced topics include management optimization and sequential data assimilation. This approach and these advanced topics assume a highly-parameterized approach, as motivated in Part 1. Topics are laid out in manner that reflects real-world workflows, with a focus on practical application of concepts and problem solving.

Part 2 is structured with a specific order for learners to follow to ensure a logical progression of topics, inline with a real-world applied workflow. Learners have the option to explore various sequences covering advanced topics, such as: - Prior Monte Carlo analysis - Highly-parameterized Gauss-Levenberg Marquardt history matching and associated Data

The Journal of Open Source Education

Worth analysis using First Order, Second Moment (FOSM) techqnique, - Ensemble-based history matching and uncertainty analysis with the iterative ensemble smoother approach as implemented in `PEST++IES`, - Sequential data assimilation with `PEST++DA`, and - Single-objective and multi-objective optimization under uncertainty with `PEST++OPT` and `PEST++MOU`.

Each of these sequences comprises multiple notebooks to be executed in a specified order. They demonstrate how to execute the workflow, interpret results, and apply the concepts to real-world problems.

In summary, the tutorial notebooks are organized to guide learners through a structured learning experience in the field of decision-support groundwater modelling. Part 0 provides foundational knowledge, while Parts 1 and 2 offer progressively advanced content. The authors attest that it is ideal to work through Parts 1 and 2 in their entirety, referring back to Part 0 for additional background. However, this amount of content requires a significant time commitment so, practically, many users will start with Part 2 and, hopefully, be able to apply the concepts to a problem of their own as they progress. Over time, referring back through Part 1 will provide a deeper understanding of some concepts and techniques taken for granted in the highly-parameterized, largely ensemble-based approaches of Part 2.

Experience of use in teaching and learning situations

The notebooks were employed during the Applied Decision Support Groundwater Modeling With Python: A Guided Self-Study Course hosted by GMDSI. This self-guided course comprised 5 online sessions, each lasting 1 to 2 hours and focused on the workflows of Part 2. During each session the instructors go through a section of the tutorials and expand on some of the concepts. Learners were tasked with going through the notebooks in between sessions to stimulate discussion and questions. Sessions were recorded and can be accessed on the GMDSI website. Beyond the live online sessions, learners were incentivized to make use of the GitHub Discussions feature to retain a search-engine findable record of common questions.

Feedback from the 65 students who participated in the course was anecdotal but informative. Figure (**(fig-responses?)**) summarizes the responses by 34 respondents to four questions, comprising 52%. The majority of respondents indicated a preference for this hybrid self-guided/online instruction approach over an in-person week-long intensive class with only one respondent indicating preference for self-guided study of the course materials only. Just under 60% of the respondents reported being able to keep up with most or all of the assigned self-study notebooks, while 41% reported falling behind. Given 5 categories of comfort level working with PEST++ (1 being most comfortable, and 5 being least) before and after the class, there was a notable shift toward higher comfort level. Interestingly, when evaluating individual responses, the majority (56%) reported being more comfortable with PEST++ after the course (defined as an increase of one level) and 15% reported being much more comfortable (an increase of two levels). However, 21% reported the same comfort level before and after while 24% reported being less or much less comfortable (a decrease or one or two levels, respectively). Without further questions, we cannot know whether these decreases reflect a humble realization that their mastery was less complete than they thought, *a priori*, or whether the material was confounding.

Open-ended feedback from the participants was generally positive and also included some constructive criticism. Participants appreciated the opportunity to ask questions and several reported hearing the discussion around other peoples' questions as being valuable and clarifying aspects of the material. The main critical suggestions included incorporating more real-world examples rather than relying, as we 100% did in the notebook design, on the synthetic model. Participants also noted the twin challenges of a large amount of information coupled with trying to be accountable to keep up in the class as potentially limiting the value relative to a week-long course. We conclude from this experience that the hybrid approach has value but there may still be a better approach for future educational opportunities.

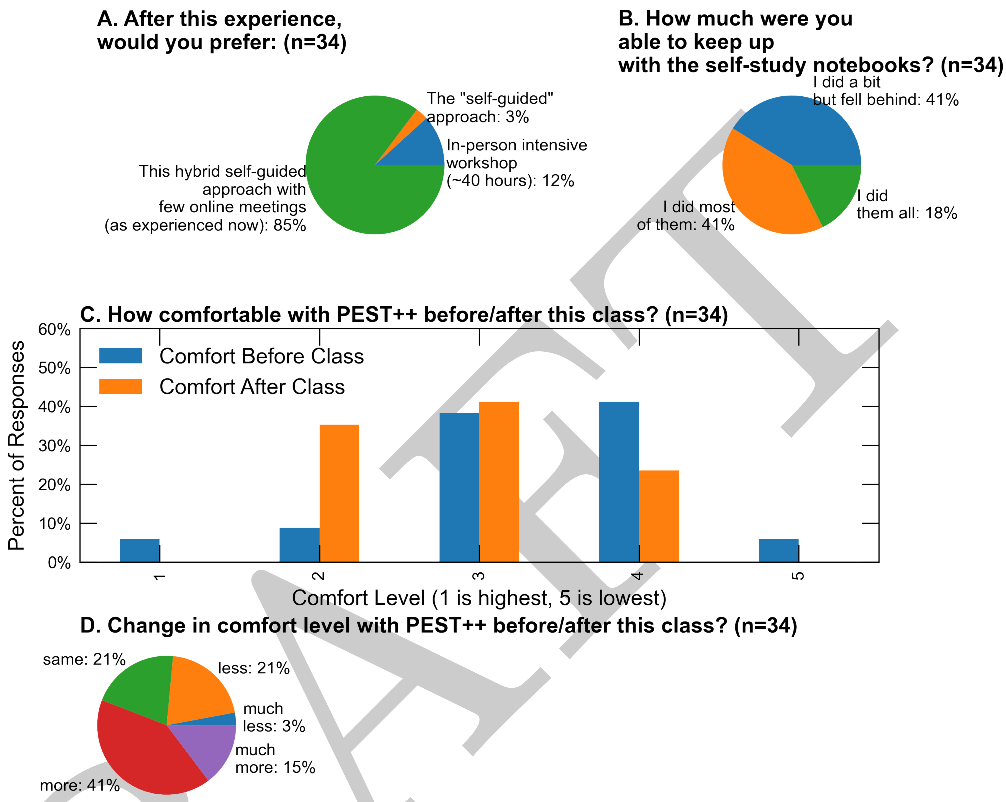

Figure 1: Summary of responses to post-course survey based on 34 responses. Panel A summarizes whether respondents would prefer and intensive in-person workshop or this hybrid option. Panel B summarizes how much of the notebooks respondents were able to complete throughout the course. Panel C summarizes respondent comfort level with PEST++ before and after the course. Panel D highlights individual changes in comfort level reported due to the course.

Acknowledgements

The tutorials were originally developed with support from the U.S Geological Survey (USGS) and support from USGS continues through the HyTest training project. Continued development and support is funded by the Groundwater Modelling Decision Support Initiative (GMDSI). GMDSI is jointly funded by BHP and Rio Tinto. We thank Dr. John Doherty for his tireless and pioneering efforts starting PEST and continuing to innovate and Dr. Randall Hunt for his leadership in PEST and PEST++ applications and development and contributions to the initial curriculum for this material and the early version of the notebooks. We finally thank users and stress-testers for their valuable feedback and continued community contributions to the repository.

References

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.

The Journal of Open Source Education

Groundwater, *54*(5), 733–739. https://doi.org/https://doi.org/10.1111/gwat.12413

Doherty, J. (2015). *PEST and Its Utility Support Software*. https://pesthomepage.org/

Fienen, M. N., & Bakker, M. (2016). HESS opinions: Repeatable research: What hydrologists can learn from the duke cancer research scandal. *Hydrology and Earth System Sciences*, *20*(9), 3739–3743. https://doi.org/10.5194/hess-20-3739-2016

Fienen, M. N., Corson-Dosch, N. T., White, J. T., Leaf, A. T., & Hunt, R. J. (2022). Risk-based wellhead protection decision support: A repeatable workflow approach. *Groundwater*, *60*(1), 71–86. https://doi.org/10.1111/gwat.13129

Glaveski, S. (2019). Where companies go wrong with learning and development. *Harvard Business Review*, *2*. https://hbr.org/2019/10/where-companies-go-wrong-with-learning-and-deve

White, Jeremy T., Fienen, M. N., & Doherty, J. E. (2016). A python framework for environmental model uncertainty analysis. *Environmental Modelling & Software*, *85*, 217–228. https://doi.org/10.1016/j.envsoft.2016.08.017

White, Jeremy T., Foster, L. K., Fienen, M. N., Knowling, M. J., Hemmings, B., & Winterle, J. R. (2020). Toward reproducible environmental modeling for decision support: A worked example. *Frontiers in Earth Science*, *8*, 50.

White, J. T., Foster, L. K., Fienen, M. N., Knowling, M. J., Hemmings, B., & Winterle, J. R. (2020). *Towards reproducible environmental modeling for decision support: A worked example*. U.S. Geological Survey data release. https://doi.org/10.5066/P9AUZMI7

White, Jeremy T., Hemmings, B., Fienen, M. N., & Knowling, M. J. (2021). Towards improved environmental modeling outcomes: Enabling low-cost access to high-dimensional, geostatistical-based decision-support analyses. *Environmental Modelling & Software*, *139*, 105022.

White, Jeremy T., Hunt, R. J., Fienen, M. N., & Doherty, J. E. (2020). *Approaches to highly parameterized inversion: PEST++ version 5, a software suite for parameter estimation, uncertainty analysis, management optimization and sensitivity analysis*. US Geological Survey Techniques; Methods 7-C26.

White, J., Fienen, M., Moore, C., & Guthke, A. (n.d.). Rapid, reproducible, and robust environmental modeling for decision support: Worked examples and open-source software tools. *Frontiers in Earth Science*, *11*, 1260581.