Penuela-Fernandez, A., Hutton, C. J., & Pianosi, F. (2021). An open-source package with interactive Jupyter Notebooks to enhance the accessibility of reservoir operations simulation and optimisation. *Environmental Modelling and Software*, *145*, Article 105188. https://doi.org/10.1016/j.envsoft.2021.105188

An open-source package with interactive Jupyter Notebooks to enhance the accessibility of reservoir operations simulation and optimisation

Andres Peñuela[1,2], Christopher Hutton[3], Francesca Pianosi[1,4]

1 - Dept. of Civil Engineering, University of Bristol, Bristol, BS8 1TR, UK

2 - Dept. of Agronomy, Unidad de Excelencia María de Maeztu (DAUCO), University of Córdoba, Edif. Da Vinci, Ctra. Madrid km 396, 14017, Córdoba, Spain

3 - Wessex Water Service Ltd, Bath, BA2 7WW, UK

4 - Cabot Institute for the Environment, University of Bristol, Bristol, BS8 1UJ, UK

Email to: apenuela@uco.es

Highlights

- We present a Python package, iRONS, for reservoir operations simulation and optimisation
- We discuss our development philosophy of modularity, minimalism, openness and accessibility
- We evaluate the use of interactive Jupyter Notebooks for documentation and training

Abstract

In this paper we present the interactive Reservoir Operations Notebooks and Software (iRONS) toolbox for reservoir modelling and optimisation. The toolbox is meant to serve the research and professional community in hydrology and water resource management and contribute to bridge the gaps between them. iRONS is composed of a package of Python core functions and a set of interactive Jupyter Notebooks. Core functions implement typical reservoir modelling tasks and the interactive Jupyter Notebooks illustrate, with practical examples, the key functionalities of iRONS. We describe our development philosophy, the key features of iRONS, and report some results of evaluating the effectiveness of interactive Jupyter Notebooks for training and knowledge transfer. The paper may be of interest also beyond the water resources management field, as an example of how Jupyter Notebooks and interactive visualisation help improving the documentation and sharing of open-source code and the communication of underpinning methodologies.

Keywords

Reservoir simulation and optimisation, Jupyter Notebooks, interactive visualisation, research software, workflows, model documentation

Software availability

• Name of Software: iRONS

• Description: The interactive Reservoir Operations Notebooks and Software (iRONS) is organised into two parts: a suite of functions (the 'Software') implementing several tasks for the simulation and optimisation of reservoir operations; and a set of Jupyter Notebooks (the 'Notebooks') that demonstrate key functionalities of the software through practical examples, and that can be run either locally or remotely via a web browser.

• Developer: Andres Peñuela (apenuela@uco.es) and Francesca Pianosi (francesca.pianosi@bristol.ac.uk)

• Funding Source: Development was partially supported by an EPSRC "Living with Environmental Uncertainty Fellowship" on "Robust and transparent planning and operation of water resource infrastructure" [EP/R007330/1].

• Source Language: Python

• Supported Systems: Windows, Mac, Linux

• License: MIT

• Availability: https://github.com/AndresPenuela/iRONS

1. Introduction

Mathematical models of reservoirs' operations can be useful computational tools to support reservoir development or re-operation studies. Such modelling tools are especially relevant nowadays given the renewed interest in dam construction for hydropower development (Zarfl et al., 2015) and in finding more environmental-friendly operations of existing dams (e.g.,Poff and Schmidt (2016); Chen and Olden (2017)). In hydrology, reservoir modelling has attracted increasing interest as one of the key steps to improve predictions of large-scale hydrological models in human-impacted river basins (Yassin et al., 2019, Dang et al., 2020, Rougé et al., 2021).

Despite an ever growing scientific literature about reservoirs operation modelling and optimisation (see e.g. reviews by Labadie (2004), Ahmad et al. (2014) and Dobson et al. (2019)) the uptake of these methods by practitioners is still very limited (Brown et al., 2015, Pianosi et al., 2020a), as is the inclusion of reservoir simulation routines – set aside optimisation ones - into large-scale hydrological models (and most recent attempts are somehow disconnected from the scientific literature on reservoir operation optimisation mentioned above; a notable exception being Turner and Galelli (2016)). We think that the availability of simple-to-use software for reservoir operations simulation and optimisation may help close these gaps. We also claim that this software should have some key characteristics: it should be modular and minimal; and it should be open and accessible.

By modularity we intend that basic tasks such as the generation of reservoir inflow time series, the determination of optimal releases from current storage or other information, etc., should be implemented in separate functions so to maximise the possibility for users to re-apply them in different contexts (Figure 1). This helps finding a coherent approach to tasks that, despite having different hydrological (e.g. short-term forecasting vs long-term prediction) or practical (day-to-day operations vs long-term planning) meaning, share a common mathematical formulation. It also facilitates the integration with other toolboxes already available for specialised tasks (e.g. optimization algorithms) so to avoid duplication of efforts whenever possible. Last, modularity enables the users to approach the software in different ways: either by plugging their tools (e.g. demand models) into the software, or by plugging-out of the software the specific components (e.g. the reservoir mass balance function or a specific reservoir operating rule) for integration into their tools (e.g. large-scale hydrological models).

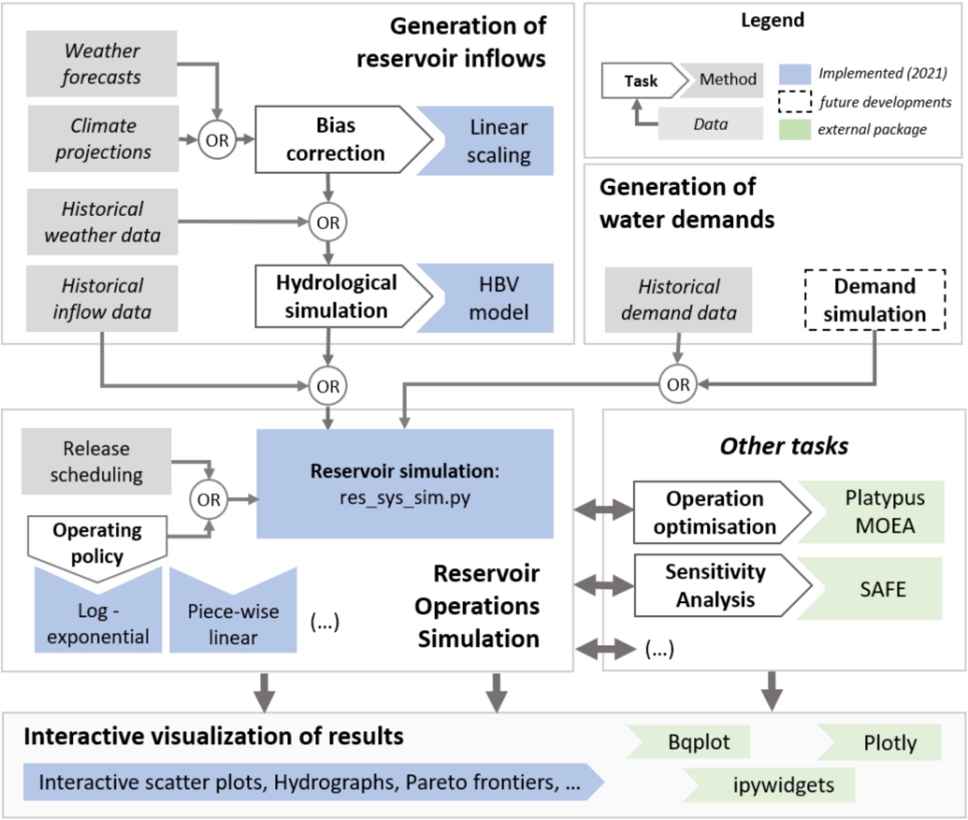


Figure 1 Modular structure of the approach used in iRONS to implement the optimisation of reservoir operation simulation and optimisation. Individual tasks can be performed according to different models or methods; the Figure highlights as an example some of those currently implemented in iRONS, integrated from other Python packages, or to be developed in future releases.

By openness we intend that, firstly, the code must be open-source, so to enable users to scrutinise its functioning and set the ground for the reproducibility and re-usability of its outputs. The lack of reproducibility of many hydrological and water resources modelling studies has been a topic of growing concern (Stagge et al., 2019, Hutton et al., 2016). Reproducibility is important both in academia, to help knowledge sharing and accumulation, and in practical settings, where transparency of decision making is required for different purposes, such as internal assurance process, regulators' approval, etc. (Badham et al., 2019). However, sharing open-source code is not sufficient without providing information for users to understand what the code does and to check whether it does it correctly when applied to other datasets or case studies. Modelling must be understandable, easy, transparent and even fun (Loucks and van Beek, 2017). This is what we mean by accessibility, which ideally should encompass users with diverse skills and level of modelling expertise.

A huge opportunity to this end comes from increasingly powerful tools to develop rich and interactive documentation in an efficient way, such as R Markdown or Jupyter Notebooks. These are literate programming environments that combine executable code, rich media, computational output and explanatory text in a single document. Through the Notebooks, users can execute the code, visualise the results, modify the code and repeat, in a kind of iterative conversation (Perkel, 2018). The result is a computational narrative that builds stronger links between model, data and results (Perez and Granger, 2007, Kluyver et al., 2016). Jupyter Notebooks have been proposed as an effective approach to implement and enhance reproducible modelling (Choi et al., 2021). Moreover, Jupyter Notebooks can be run on the cloud by using online platforms, such as Binder (https://mybinder.org/), so that they are accessible by a web browser without requiring the installation of Python.

Another opportunity provided by Notebooks, in an enhancement to static documentation, is the possibility to include interactive visualization libraries such as Bqplot (https://bqplot.readthedocs.io/) and Plotly (https://plotly.com/) and implement reactive plots such as markers and intuitive elements such as sliders within the figures of a Jupyter Notebook. When the user interacts with one of these interactive plots or elements, for instance by mouse clicking or dragging, all the relevant figures are automatically updated. Interactive visualisation is key to enhance the accessibility and value of model predictions, to facilitate the communication of results beyond traditional static documents, and ultimately to support complex decision-making problems (Woodruff et al., 2013). As water resource management becomes increasingly multi-dimensional and multi-objective, interactive visualisation has been proposed in particular to enable the exploration of tradeoffs between multiple objectives, e.g. via interactive multi-dimensional Pareto fronts and parallel coordinate plots (Kwakkel, 2017, Hadka et al., 2015, Kasprzyk et al., 2013, Yang et al., 2016, Matrosov et al., 2015). However, the effectiveness of these visualisation tools may diminish with increasing number of dimensions (Parker et al., 2015). The use of more intuitive elements such as sliders or coloured 2D scatter plots may provide a more effective mechanism to communicate complex modelling concepts to a non-technical audience such as decision-makers and stakeholders within and outside water companies.

In this paper we present the interactive Reservoir Operation Notebooks and Software (iRONS), a Python toolbox that implements several functions for the simulation and optimisation of reservoir operations, and is based on the above principles of modularity, minimalism, openness and accessibility. The target users of iRONS are students, researchers and practitioners that need a simple to use, and yet rigorous, Python code for reservoir simulation and/or optimisation – be it for supporting short-term operational decisions, long-term reservoir operation planning, or to simulate dammed river basins. Compared to other available packages for reservoir simulation and optimisation, such as the web-based application ResOS (Jahanpour et al., 2014) or the R package *reservoir* (Turner and Galelli, 2016), iRONS has the advantage of combining the user-friendliness of web-based applications, thanks to the interactive Jupyter Notebooks, with the transparency and adaptability of the Python functions.

Our paper may also be of interest for modellers and developers of research software beyond the reservoir operation application field, as an illustration of ways by which Jupyter Notebooks and interactive visualisation can contribute to improve the documentation and sharing of open-source code and the communication of underpinning methodologies. It also provides a demonstration of the potential of interactive Jupyter Notebooks for training and knowledge transfer from more to less specialised user groups. Given the ease with which interactive Jupyter Notebooks can be developed, they can potentially revolutionise the documentation and sharing of research software towards an unprecedented level of transparency for relatively low effort (at least for sufficiently 'parsimonious' models – i.e. models that can be run on a desktop computer or on a cloud server in a matter of seconds/minutes).

The reminder of the paper is organised as follows. In the next Section, we briefly present the structure and characteristics of the two key parts of iRONS: the software and the Notebooks. We then present the results of some evaluation experiments we ran with students and early-career researchers to evaluate the effectiveness of Jupyter Notebooks for knowledge transfer.

2. Structure, content and style

The interactive Reservoir Operations Notebooks and Software (iRONS) is organised into two parts: a suite of functions (the 'Software') implementing several tasks related to the simulation and optimisation of reservoir operations; and a set of Jupyter Notebooks (the 'Notebooks') that demonstrate key functionalities of the software through practical examples, and that can be run either locally or remotely via a web browser.

2.1 The Software

The software can be used to simulate and/or support reservoir operations over the short-term, i.e. to optimise the reservoir' release scheduling over the coming days/weeks, or in the long-term, i.e. through the optimisation and evaluation of the reservoir's operating policy (Dobson et al., 2019). Thanks to its modular structure, any of these components can be easily linked/integrated into other water resource system software, such as Pywr (Tomlinson et al., 2020).

Few different options are available for the shape of the operating policy, including a piece-wise linear function and a log-exponential function (Rougé et al., 2021, Proussevitch et al., 2013) – but more can be easily integrated. Optimisation of both (short-term) release scheduling and (long-term) operating policy is performed by linking iRONS to the Platypus Multi-Objective Evolutionary Algorithm (MOEA) (https://github.com/Project-Platypus/Platypus). In the former case the decision variables optimised by Platypus MOEA are the reservoir releases, whereas in the latter case the decision variables to be optimised are the parameters of the operating policy (this is the so called "policy search" or "parameterisation-simulation-optimisation" approach to reservoir optimisation, see discussion in Dobson et al. (2019)). The current version of iRONS also includes the Python implementation of a lumped hydrological model, the rainfall-runoff HBV model (Bergström and Singh, 1995), which is used in several Notebooks to show the modelling chain from meteorological inputs to reservoir inflows. However, depending on the application, the user may want instead to import a time series of inflow records or inflows generated elsewhere by a different model.

Table 1 - list and description of the functions implemented in the iRONS software at the time of writing this paper.

Generation of inflow forecasts

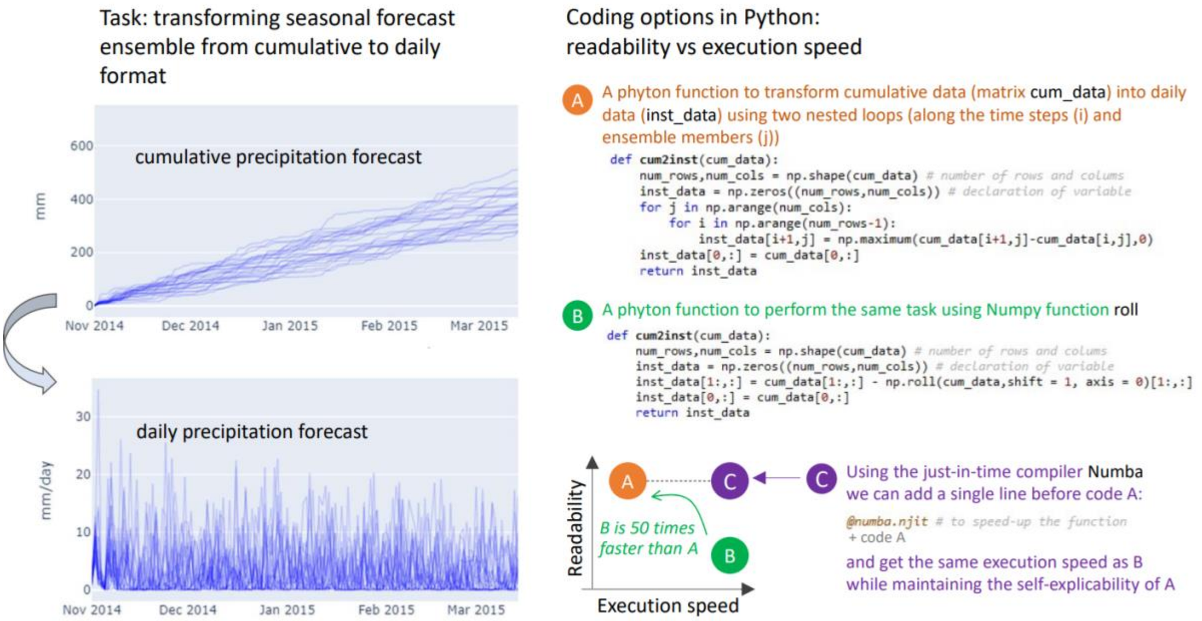
File	Function	Description
download_forecast.py	data_retrieval_request	downloads ECMWF seasonal forecast data
read_data.py	read_csv_data	extracts the data from a CSV file
	read_netcdf_data	extracts the data from a NetCDF file
cum2inst.py	cum2inst	transforms cumulative data into instantaneous data
bias_correction.py	linear_scaling	bias corrects meteorological data according to a linear scaling approach
HBV_sim.py	HBV_sim	simulates catchment-scale rainfall-runoff processes according to the HBV model by Bergström and Singh (1995)
HBV_calibration.py	HBV_calibration	calibrates the HBV model
day2week2month.py	day2week	transforms daily data into weekly data
	day2month	transforms daily data into monthly data

Reservoir operations simulation

File	Function	Description
res_sys_sim.py	mass_bal_func	closes the reservoir water balance equation
	res_sys_sim	makes operating policies understandable by mass_bal_func and compatible with Numba
operating_policy.py	op_piecewiselin_1res	calculates reservoir release from storage based on a piece-wise operating policy
	op_logexp_1res_v1	calculates reservoir release from storage on the log-exp policy proposed by Proussevitch et al. (2013)
	op_logexp_1res_v2	calculates reservoir release from storage based on the log-exp policy proposed by Rougé et al. (2021)

Table 1 provides the list of functions implemented in the iRONS software at the time of writing. Our plan is to keep adding to the list, particularly including other operating policy functions for different reservoir purposes and configurations, and possibly implementing some simple demand models. As anticipated in the Introduction, we would focus future efforts on the reservoir modelling aspects while relying on integration of other packages for more sophisticated demand and inflow predictions - for example, the Python package for catchment-scale rainfall-runoff modelling SuperflexPy (Dal Molin et al., 2020) - in order to avoid unnecessary duplication of efforts. Another area of further development could be the linking to other optimisation tools beyond Platypus MOEA. However, we shall highlight that comparing different optimisation techniques is a technical task, probably of interest mainly to algorithm developers who are well equipped to develop their own code, and as such is somehow beyond the scope of iRONS – which instead aims to make optimisation (as implemented in some standard algorithms) accessible to the non-specialist user.

While Python is an object-oriented language, iRONS uses a functional programming style which, in our opinion, requires lower coding skills to be read and (if needed) revised and which has been successfully applied by widely adopted toolboxes such as SAFE (Pianosi et al., 2020b). Moreover, whenever possible, the code is written in "math-like" style aiming to maximise readability, occasionally using less efficient but more legible coding options (for example, a for loop in place of vectorisation). Computational efficiency is regained by making the code compatible with the Numba just-in-time compiler (http://numba.pydata.org/; Lam et al. (2015); Marowka (2018)) designed to work best on code that uses loops and Numpy arrays and functions so that "math-like" functions can still be used within computationally expensive tasks. An illustration of the concept is given in Figure *2*. As a matter of example of the computational requirements of iRONs, the real-time optimisation of the 2-reservoirs system presented in Peñuela et al. (2020), which used 2,500,000 executions of the res_sys_sim function over a 21 weeks simulation period (this figure stems from computing average reservoir performance against an ensemble of 25 members, and allowing 100,000 function evaluations by the MOEA), required 80 seconds on a standard desktop computer (Intel Core i7-6700 CPU @ 3.40GHz).


Task: transforming seasonal forecast ensemble from cumulative to daily format

Coding options in Python: readability vs execution speed

A A phyton function to transform cumulative data (matrix cum_data) into daily data (inst_data) using two nested loops (along the time steps (i) and ensemble members (j))

```
def cum2inst(cum_data):
    num_rows,num_cols = np.shape(cum_data) # number of rows and colums
    inst_data = np.zeros((num_rows,num_cols)) # declaration of variable
    for j in np.arange(num_cols):
        for i in np.arange(num_rows-1):
            inst_data[i+1,j] = np.maximum(cum_data[i+1,j]-cum_data[i,j],0)
    inst_data[0,:] = cum_data[0,:]
    return inst_data
```

B A phyton function to perform the same task using Numpy function roll

```
def cum2inst(cum_data):
    num_rows,num_cols = np.shape(cum_data) # number of rows and colums
    inst_data = np.zeros((num_rows,num_cols)) # declaration of variable
    inst_data[1:,:] = cum_data[1:,:] - np.roll(cum_data,shift = 1, axis = 0)[1:,:]
    inst_data[0,:] = cum_data[0,:]
    return inst_data
```

C Using the just-in-time compiler Numba we can add a single line before code A:

```
@numba.njit # to speed-up the function
+ code A
```

and get the same execution speed as B while maintaining the self-explicability of A

Figure 2 Example of different coding options for the task of transforming cumulative data (such as the ensemble seasonal precipitation forecast released by the European Centre for Medium-Range Weather Forecasts, ECMWF) into daily data (as used for reservoir daily mass balance equation). Using the just-in-time compiler Numba (option C) allows to mitigate the tradeoff between pursuing readability (option A) and speed of execution (option B).

2.2 The Notebooks

The Jupyter Notebooks are divided in two sections:

1) Knowledge Transfer Notebooks: a set of simple examples to demonstrate the value of simulation and optimisation tools for reservoir operations by application to 'proof-of-concept' systems. The Notebooks cover a range of concepts relevant to reservoir operation, such as: manual vs automatic calibration of rainfall-runoff models used to generate reservoir inflows; what-if analysis vs optimisation of reservoir releases; optimisation under conflicting objectives and under uncertainty; optimisation of release releases scheduling vs optimisation of an operating policy; different shapes of operating policies for different reservoir purposes such as domestic or irrigation supply, flood control, or hydropower production.

2) Implementation Notebooks: a set of workflow examples showing how to apply the iRONS functions to real-world data and problems including: generating inflow forecasts through a rainfall-runoff model, including bias correcting weather forecasts; optimising release scheduling against an inflow scenario or a forecast ensemble; optimising an operating policy against time series of historical or synthetic inflows. These Notebooks are meant to serve as a 'learn-by-doing' alternative to a User manual and a starting point for the user's own application workflows (Pianosi et al., 2020b).The intended learning objective of the two types of notebooks and their key features are summarised in Table 2. The list and description of all the Notebooks included in iRONS at the time of writing is given in Table 3. Some examples of interactive visualisations included in the Knowledge Transfer Notebooks are given in Figure 3. These and other visualizations were implemented using integrated widgets (ipywidgets; https://ipywidgets.readthedocs.io/) and the interactive visualization libraries Bqplot (https://bqplot.readthedocs.io/) and Plotly (https://plotly.com/).

Table 2 - intended objectives and key features of the interactive Jupyter Notebooks

Knowledge Transfer (KT) Notebooks	**Implementation Notebooks**
Objective: familiarise non-specialists with key concepts of reservoir operations simulation and optimisation	*Objective*: illustrate how to implement reservoir operations simulation and optimisation tasks using iRONs
Key features: 1) Simplified case study 2) Only most salient pieces of code shown 3) Interactive visualisation to stimulate user engagement (for example, sliders to change variable values, such as weekly reservoir release scheduling, in order to 'beat' a specified performance target)	*Key features:* 1) Real-world (or realistic) case studies 2) Full code shown and explained step by step 3) Interactive visualisation to enable increasingly in-depth exploration of the results (for example, click-drag interactions such as zoom, pan, select, or mouseover to display additional information about a specific datapoint)

3. Evaluation of interactive Jupyter Notebooks

To our knowledge, the efficacy of interactive Jupyter Notebooks for knowledge transfer has been claimed but never actually measured. Previous studies have looked at the efficacy of interactive visualisation in teaching settings, coming to mixed conclusions (Chou, 2003). For example, Chien and Chang (2012) and Akinlofa et al. (2013) both found that full interactions with visualisation tools was more effective for learning than watching (non-interactive) animations of videos. However, Pedra et al. (2015) showed that incorporating sophisticated interactivity features into lessons increased the interest of students, but this was not translated into better learning outcomes. In this section we present the results of some experiments we ran to measure the efficacy of our Knowledge Transfer (KT) Notebooks.

Table 3 - list and description of the Jupyter Notebooks included in iRONS at the time of writing this paper

Knowledge Transfer Notebooks

Name	Description
1.a. Simple example of how to use Jupyter Notebooks	uses a sound wave example to show how to use Jupyter Notebooks and interactive visualization elements
2.a. Calibration and evaluation of a rainfall-runoff model	shows how to calibrate, both manually and automatically, a rainfall-runoff model, which could be used to estimate reservoir inflows
3.a. Recursive decisions and multi-objective optimisation	shows how simulation models and optimisation algorithms can be used to derive optimal release scheduling for a simple water reservoir system
3.b. Decision making under uncertainty	shows how to deal with uncertainty - and the difference between deterministic and stochastic optimisation - when deriving optimal reservoir releases
3.c. Reservoir operating policy	shows how to optimise the operating policy of a water reservoir system (using a “policy search” approach)
3.d. Types of reservoir operating policies	shows how to define different operating policies for different reservoir purposes and how to account for variability over the year

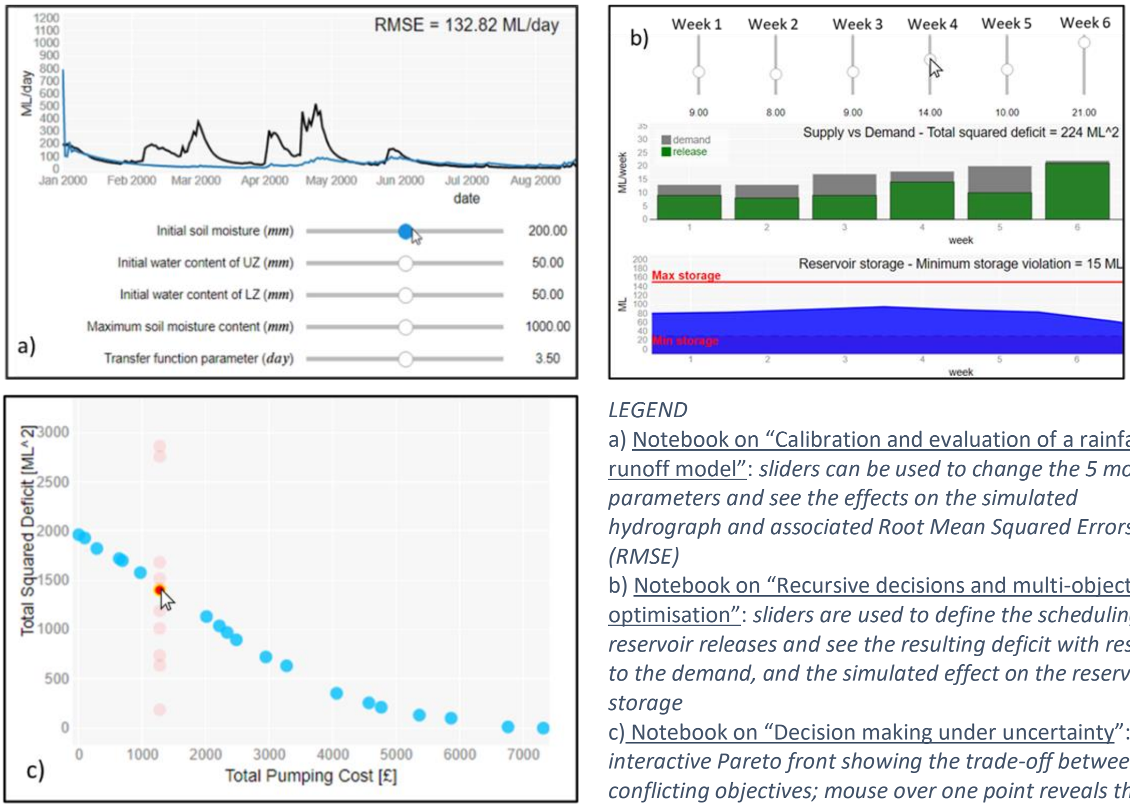
Implementation Notebooks

1.a. Downloading ensemble weather forecasts	downloads ensemble weather forecasts/hindcasts from the ECMWF public dataset
1.b. Bias correction of weather forecasts	applies a linear scaling approach to correct the bias of seasonal forecasts (rainfall, temperature and evaporation ensembles)
2.a. Generation of reservoir inflow ensemble forecasts from weather forecasts	uses the HBV model to simulate the inflow forecast into a water reservoir using the seasonal forecasts, both in a deterministic and in a stochastic manner, as forcing inputs
3.a. Multi-objective optimisation of reservoir release scheduling	simulates and applies multi-objective optimization to find optimal release and pumped inflows scheduling in some illustrative study-cases
3.b. Multi-objective optimisation of reservoir operating policy	uses an operating policy function, together with a simulation model and optimisation algorithm, to assist the long-term operation of a water reservoir system

Specifically, we shared two of the Knowledge Transfer (KT) Notebooks available in iRONS (the one on “Calibration and evaluation of a rainfall-runoff model” and the one on “Recursive decisions and multi-objective optimisation”) during online workshops or after online talks, and asked the participants to go through them at their pace. Before and after going through the KT Notebooks, participants filled in a questionnaire to measure their familiarity and understanding of the concepts covered by the KT Notebooks. By comparing responses before and after, we can thus assess whether their knowledge was increased by using the Notebooks. Overall, we ran one in-person and two online workshops for a total of 25 participants, and obtained responses by email from another 7 participants - consisting of 26 PhD students with diverse backgrounds (13 of them in Hydrology and Water Informatics, 6 in Agronomy, 5 in Biogeochemical flows and 2 in Fine Chemistry) and 6 master’s degree students (all of them in Environmental Hydraulics). While this number of participants is limited, we think the results are nonetheless worth sharing, as is our assessment approach in the first place.

Familiarity to a concept is self-assessed by the participants using a scale from 1 (“not confident”) to 5 (“very confident”). Understanding of a concept is instead measured by the ability to give the correct answer to a close-ended question about that concept. The survey also includes direct questions to the participants about their opinion of our interactive Notebooks compared to other learning methods such as lecture slides, textbooks or online videos.

The majority of participants (72%) said that the interactive Notebooks are “better” than other learning methods that they had previously used and found very effective, about a third (28%) responded that they are “as good” and no participant said they are “worse”. “Code explanations”, “Interactivity” and “step-by-step structure” were among the reasons most commonly given for the better effectiveness for learning. Figure 4 reports the average increase in familiarity (x axis) against the increase in understanding (y axis) after engaging with the Notebooks. In general, the participants increased both their familiarity and understanding of the different technical concepts covered by the Notebooks, however there is not necessarily a clear relationship between the two. A possible explanation is that most participants felt considerably more confident initially about some of the concepts covered in the KT Notebooks than others. For instance, the initial average familiarity of the participants to the concepts of calibration and evaluation of hydrological models was 2.8 and 3.0 respectively, while for the concepts of reservoir' hedging rules and trade-off/Pareto front it was 1.6 and 2.0 respectively. As a consequence, the room for improvement in familiarity is smaller for some of the concepts and the increase of familiarity with respect to the increase of understanding does not correspond in the same way.



LEGEND

a) Notebook on "Calibration and evaluation of a rainfall-runoff model": *sliders can be used to change the 5 model parameters and see the effects on the simulated hydrograph and associated Root Mean Squared Errors (RMSE)*

b) Notebook on "Recursive decisions and multi-objective optimisation": *sliders are used to define the scheduling of reservoir releases and see the resulting deficit with respect to the demand, and the simulated effect on the reservoir storage*

c) Notebook on "Decision making under uncertainty": *interactive Pareto front showing the trade-off between conflicting objectives; mouse over one point reveals the uncertainty of the simulated performance due to inflow forecast uncertainty*

Figure 3 Examples of interactive visualisations included in the Knowledge Transfer Notebooks.

In the future we hope to be able to run more similar workshops with a more diverse range of users so to expand our understanding of the benefits – and limitations - of using Notebooks for model documentation and training.

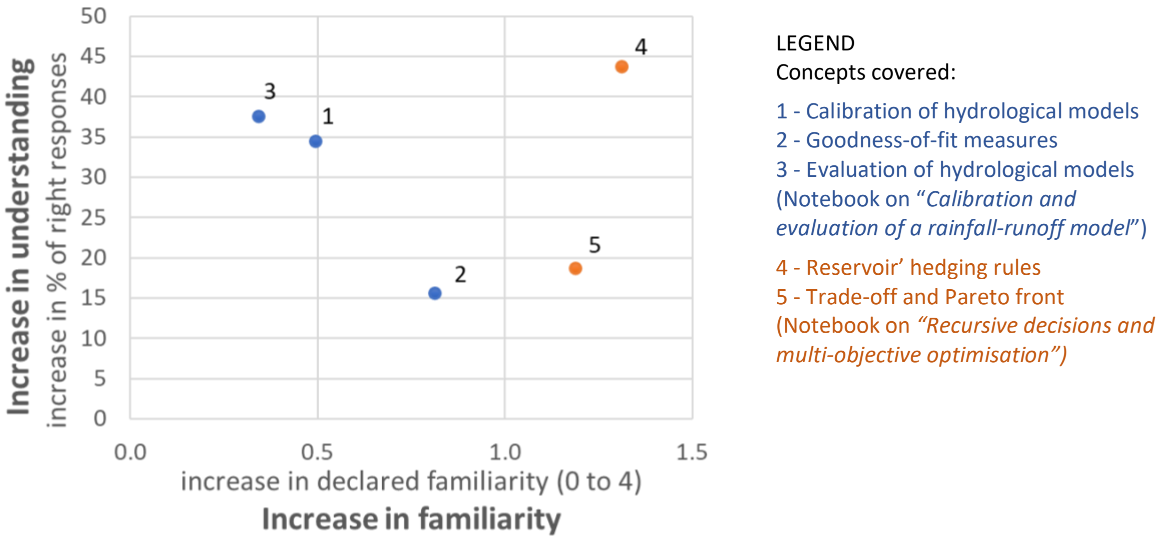


Figure 4 Increase in understand and familiarity to some key concepts after engaging with the interactive KT Notebooks. See the appendix for details of the questions asked to measure familiarity and understanding.

4. Conclusions

In this paper we presented our interactive Reservoir Operations Notebooks and Software (iRONS) toolbox for reservoir modelling and optimisation. We illustrated our development philosophy and presented some results of our first attempts at evaluating the effectiveness of interactive Jupyter Notebooks for knowledge transfer. These results suggest that the characteristics of the Notebooks (literate programming and step-by-step structure) combined with visual interactivity do enhance learning and are appreciated by users.

Future work will move along two directions. First, we plan to expand the range of core functions of iRONS, for instance by implementing more operating policy functions, or simple demand simulation models. Second, we plan to further evaluate the efficacy of the interactive Jupyter Notebooks particularly as a knowledge transfer tool for practitioners - including water resource managers, consultants and regulators.

Acknowledgements

This work was funded by the Engineering and Physical Sciences Research Council (EPSRC), grant EP/R007330/1. The authors are also grateful to the workshop participants who contributed to the evaluation of interactive Jupyter Notebooks.

References

AHMAD, A., EL-SHAFIE, A., RAZALI, S. F. M. & MOHAMAD, Z. S. 2014. Reservoir Optimization in Water Resources: a Review. *Water Resources Management,* 28**,** 3391-3405.

AKINLOFA, O. R., HOLT, P. O. B. & ELYAN, E. 2013. Domain expertise and the effectiveness of dynamic simulator interfaces in the acquisition of procedural motor skills. *British Journal of Educational Technology,* 44**,** 810-820.

BADHAM, J., ELSAWAH, S., GUILLAUME, J. H. A., HAMILTON, S. H., HUNT, R. J., JAKEMAN, A. J., PIERCE, S. A., SNOW, V. O., BABBAR-SEBENS, M., FU, B., GOBER, P., HILL, M. C., IWANAGA, T., LOUCKS, D. P., MERRITT, W. S., PECKHAM, S. D., RICHMOND, A. K., ZARE, F., AMES, D. & BAMMER, G. 2019. Effective modeling for Integrated Water Resource Management: A guide to contextual practices by phases and steps and future opportunities. *Environmental Modelling & Software,* 116**,** 40-56.

BERGSTRÖM, S. & SINGH, V. 1995. The HBV model. *Computer models of watershed hydrology.*, 443-476.

BROWN, C. M., LUND, J. R., CAI, X., REED, P. M., ZAGONA, E. A., OSTFELD, A., HALL, J., CHARACKLIS, G. W., YU, W. & BREKKE, L. 2015. The future of water resources systems analysis: Toward a scientific framework for sustainable water management. *Water Resources Research,* 51, 6110-6124.

CHEN, W. & OLDEN, J. D. 2017. Designing flows to resolve human and environmental water needs in a dam-regulated river. *Nature Communications,* 8, 2158.

CHIEN, Y.-T. & CHANG, C.-Y. 2012. Comparison of Different Instructional Multimedia Designs for Improving Student Science-Process Skill Learning. *Journal of Science Education and Technology,* 21, 106-113.

CHOI, Y.-D., GOODALL, J. L., SADLER, J. M., CASTRONOVA, A. M., BENNETT, A., LI, Z., NIJSSEN, B., WANG, S., CLARK, M. P., AMES, D. P., HORSBURGH, J. S., YI, H., BANDARAGODA, C., SEUL, M., HOOPER, R. & TARBOTON, D. G. 2021. Toward open and reproducible environmental modeling by integrating online data repositories, computational environments, and model Application Programming Interfaces. *Environmental Modelling & Software,* 135, 104888.

CHOU, C. 2003. Interactivity and interactive functions in web-based learning systems: a technical framework for designers. *British Journal of Educational Technology,* 34, 265-279.

DAL MOLIN, M., KAVETSKI, D. & FENICIA, F. 2020. SuperflexPy 1.2.0: an open source Python framework for building, testing and improving conceptual hydrological models. *Geosci. Model Dev. Discuss.,* 2020, 1-39.

DANG, T. D., CHOWDHURY, A. F. M. K. & GALELLI, S. 2020. On the representation of water reservoir storage and operations in large-scale hydrological models: implications on model parameterization and climate change impact assessments. *Hydrol. Earth Syst. Sci.,* 24, 397-416.

DOBSON, B., WAGENER, T. & PIANOSI, F. 2019. An argument-driven classification and comparison of reservoir operation optimization methods. *Advances in Water Resources,* 128, 74-86.

HADKA, D., HERMAN, J., REED, P. & KELLER, K. 2015. An open source framework for many-objective robust decision making. *Environmental Modelling & Software,* 74, 114-129.

HUTTON, C., WAGENER, T., FREER, J., HAN, D., DUFFY, C. & ARHEIMER, B. 2016. Most computational hydrology is not reproducible, so is it really science? *Water Resources Research,* 52, 7548-7555.

JAHANPOUR, M., SANDOVAL‐SOLIS, S. & AFSHAR, A. J. J. A. W. W. A. 2014. A web‐based application for optimization of single reservoir operation. 106, E509-E517.

KASPRZYK, J. R., NATARAJ, S., REED, P. M. & LEMPERT, R. J. 2013. Many objective robust decision making for complex environmental systems undergoing change. *Environmental Modelling & Software,* 42, 55-71.

KLUYVER, T., RAGAN-KELLEY, B., PÉREZ, F., GRANGER, B. E., BUSSONNIER, M., FREDERIC, J., KELLEY, K., HAMRICK, J. B., GROUT, J. & CORLAY, S. Jupyter Notebooks-a publishing format for reproducible computational workflows. ELPUB, 2016. 87-90.

KWAKKEL, J. H. 2017. The Exploratory Modeling Workbench: An open source toolkit for exploratory modeling, scenario discovery, and (multi-objective) robust decision making. *Environmental Modelling & Software,* 96, 239-250.

LABADIE, J. W. 2004. Optimal Operation of Multireservoir Systems: State-of-the-Art Review. *Journal of Water Resources Planning and Management,* 130, 93-111.

LAM, S. K., PITROU, A. & SEIBERT, S. 2015. Numba: a LLVM-based Python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC.* Austin, Texas: ACM.

LOUCKS, D. P. & VAN BEEK, E. 2017. Water Resource Systems Modeling: Its Role in Planning and Management. *Water Resource Systems Planning and Management: An Introduction to Methods, Models, and Applications.* Cham: Springer International Publishing.

MAROWKA, A. 2018. Python accelerators for high-performance computing. *The Journal of Supercomputing,* 74**,** 1449-1460.

MATROSOV, E. S., HUSKOVA, I., KASPRZYK, J. R., HAROU, J. J., LAMBERT, C. & REED, P. M. 2015. Many-objective optimization and visual analytics reveal key trade-offs for London's water supply. *Journal of Hydrology,* 531**,** 1040-1053.

PARKER, A. M., SRINIVASAN, S. V., LEMPERT, R. J. & BERRY, S. H. 2015. Evaluating simulation-derived scenarios for effective decision support. *Technological Forecasting and Social Change,* 91**,** 64-77.

PEDRA, A., MAYER, R. E. & ALBERTIN, A. L. 2015. Role of Interactivity in Learning from Engineering Animations. *Applied Cognitive Psychology,* 29**,** 614-620.

PEÑUELA, A., HUTTON, C. & PIANOSI, F. 2020. Assessing the value of seasonal hydrological forecasts for improving water resource management: insights from a pilot application in the UK. *Hydrol. Earth Syst. Sci. Discuss.,* 2020**,** 1-26.

PEREZ, F. & GRANGER, B. E. 2007. IPython: A System for Interactive Scientific Computing. *Computing in Science & Engineering,* 9**,** 21-29.

PERKEL, J. M. 2018. Why Jupyter is data scientists' computational notebook of choice. *Nature,* 563**,** 145-147.

PIANOSI, F., DOBSON, B. & WAGENER, T. 2020a. Use of Reservoir Operation Optimization Methods in Practice: Insights from a Survey of Water Resource Managers. *Journal of Water Resources Planning and Management,* 146**,** 02520005.

PIANOSI, F., SARRAZIN, F. & WAGENER, T. 2020b. How successfully is open-source research software adopted? Results and implications of surveying the users of a sensitivity analysis toolbox. *Environmental Modelling & Software,* 124**,** 104579.

POFF, N. L. & SCHMIDT, J. C. 2016. How dams can go with the flow. *Science,* 353**,** 1099-1100.

PROUSSEVITCH, A., SHIKLOMANOV, A., FROLKING, S., GLIDDEN, S., LAMMERS, R. B. & WISSER, D. Log-Exponential Reservoir Operating Rules for Global and Regional Hydrological Modeling. AGU Fall Meeting Abstracts, 2013. GC21B-0827.

ROUGÉ, C., REED, P. M., GROGAN, D. S., ZUIDEMA, S., PRUSEVICH, A., GLIDDEN, S., LAMONTAGNE, J. R. & LAMMERS, R. B. 2021. Coordination and control – limits in standard representations of multi-reservoir operations in hydrological modeling. *Hydrol. Earth Syst. Sci.,* 25**,** 1365-1388.

STAGGE, J. H., ROSENBERG, D. E., ABDALLAH, A. M., AKBAR, H., ATTALLAH, N. A. & JAMES, R. 2019. Assessing data availability and research reproducibility in hydrology and water resources. *Scientific Data,* 6**,** 190030.

TOMLINSON, J. E., ARNOTT, J. H. & HAROU, J. J. 2020. A water resource simulator in Python. *Environmental Modelling & Software,* 126**,** 104635.

TURNER, S. W. D. & GALELLI, S. 2016. Water supply sensitivity to climate change: An R package for implementing reservoir storage analysis in global and regional impact studies. *Environmental Modelling & Software,* 76**,** 13-19.

WOODRUFF, M. J., REED, P. M. & SIMPSON, T. W. 2013. Many objective visual analytics: rethinking the design of complex engineered systems. *Structural and Multidisciplinary Optimization,* 48**,** 201-219.

YANG, Y. C. E., WI, S., RAY, P. A., BROWN, C. M. & KHALIL, A. F. 2016. The future nexus of the Brahmaputra River Basin: Climate, water, energy and food trajectories. *Global Environmental Change,* 37**,** 16-30.

YASSIN, F., RAZAVI, S., ELSHAMY, M., DAVISON, B., SAPRIZA-AZURI, G. & WHEATER, H. 2019. Representation and improved parameterization of reservoir operation in hydrological and land-surface models. *Hydrol. Earth Syst. Sci.,* 23**,** 3735-3764.

ZARFL, C., LUMSDON, A. E., BERLEKAMP, J., TYDECKS, L. & TOCKNER, K. 2015. A global boom in hydropower dam construction. *Aquatic Sciences,* 77**,** 161-170.

Appendix: questionnaire (with responses) for assessing the effectiveness of interactive Notebooks

Questions and their answers before and after engaging with the Notebook from the 32 students that participated in our workshops or email surveys.

Questions for self-assessment of familiarity		
1. How familiar do you feel with the concept of "Calibration of a hydrological model"		
	BEFORE (% of participants)	AFTER (% of participants)
1 (not confident)	19	9
2	22	16
3	31	25
4	22	41
5 (very confident)	6	9
2. How familiar do you feel with the concept of "Goodness-of-fit measures"		
	BEFORE (%)	AFTER (%)
1 (not confident)	19	6
2	22	13
3	38	19
4	13	47
5 (very confident)	9	16
3. How familiar do you feel with the concept of "Validation of a hydrological model"		
	BEFORE (%)	AFTER (%)
1 (not confident)	9	6
2	25	22
3	34	19
4	19	38
5 (very confident)	13	16
4. How familiar do you feel with the concept of "Hedging rules"		
	BEFORE (%)	AFTER (%)
1 (not confident)	59	13
2	28	28
3	9	19
4	3	41
5 (very confident)	0	0
5. How familiar do you feel with the concept of "Trade-off and Pareto front"		
	BEFORE (%)	AFTER (%)
1 (not confident)	50	13
2	16	16
3	25	25
4	6	38
5 (very confident)	3	9

Questions for measuring understanding (correct answer in bold)		
1. Does the influence of a given parameter on the simulated hydrograph depends on the value of the other parameters too?		
	BEFORE (%)	AFTER (%)
Yes	**53**	**88**
No	9	3
Not sure / I don´t know	38	9
2. What errors have a stronger influence on the Root-mean-square-error (RMSE) value?		
	BEFORE (%)	AFTER (%)
Errors of high flow predictions	**38**	**53**
Errors in the prediction during prolonged dry events	13	38
Not sure / I don´t know	50	9
3. How could you increase the robustness of the calibration results, so the optimal set of parameters is more likely to be valid for future years?		
	BEFORE (%)	AFTER (%)
By increasing the number of iterations in the calibration process	16	28
By increasing the number of years used for the calibration	**34**	**72**
Not sure / I don´t know	50	0
4. What is the main purpose of a hedging rule for reservoir operation?		
	BEFORE (%)	AFTER (%)
Reduce the probability of a severe water shortage	**44**	**88**
Reduce the operation costs	0	3
Reduce the river abstractions	6	3
Not sure / I don´t know	50	6
5. What is the optimal release policy of a system with two objectives: resource conservation and reliability of water supply?		
	BEFORE (%)	AFTER (%)
The policy that prioritizes the minimization of supply deficits	16	25
The policy that prioritizes the maximization of the storage level	6	3
There is not a single optimal solution	**44**	**63**
Not sure / I don´t know	34	9