25th International Congress on Modelling and Simulation, Darwin, NT, Australia, 9 to 14 July 2023
mssanz.org.au/modsim2023

Design and implementation of a software tool supporting the Inter-Provincial Water Apportionment Accord in Pakistan

J.-M. Perraud [a], **A.C. Freebairn** [a], **S.P. Seaton** [a], **Y. Yu** [c], **G.M. Podger** [b], **M.D. Ahmad** [a] **and S.M. Cuddy** [a]

[a] *CSIRO Environment, Canberra, ACT*
[b] *Water Bublu Pty Ltd, Canberra, ACT*
[c] *CSIRO Environment, Brisbane, QLD*
Email: jean-michel.perraud@csiro.au

Abstract: The Indus Basin Irrigation System (IBIS) in Pakistan is the largest contiguous irrigation system. The weather cycle is characterised by a monsoonal regime with wet summer (Kharif) and dry winter (Rabi) seasons. The high-level pre-seasonal planning of the operations of the two largest reservoirs of Pakistan, Tarbela (~7.4 km3 in 2018) and Mangla (~9 km3 in 2020), is undertaken twice a year in March and September by federal and provincial water agencies. The sharing of water between provinces is governed by the principles in the 1991 Inter-Provincial Water Apportionment Accord (WAA). Interpreting and implementing these principles is a complicated process understood by few experts. Water sharing among provinces is a contentious topic, and the trust of the stakeholders in the transparency, neutrality and consistency of the procedures and decision tools is very important. In close partnership with stakeholders in Pakistan, the CSIRO has undertaken to help elicit and document this expert knowledge. One recent output of this work is a modern software application, the WAA Tool. After iterative co-design and testing over several seasonal planning cycles, the application has been adopted by stakeholders as a neutral implementation that matches the historically established procedures.

The WAA Tool is a software product built with the mainstream languages Python and Javascript and their ecosystem of libraries, to facilitate ongoing maintenance and evolutions. It is currently a desktop application to support the existing practices but is designed to evolve in the future to a web-based application if required by changes in the systems used by stakeholder water authorities and departments in Pakistan. The model algorithms are available as a Python package, and exploratory analysis of the reservoir operations and water sharing can now be undertaken from interactive Jupyter notebooks, opening new logistical possibilities for the study of the IBIS. The WAA tool is a reimplementation of a proof-of-concept application written in Fortran. This Fortran application was *de facto* numerically a reference implementation that needed to be exactly matched by the new implementation. Given the intricate nature of the algorithms, the Fortran code was reengineered into a dynamic library accessed via a Python wrapper, to enable tests at a smaller granularity. This approach also offered a fallback position in case porting to pure Python could not be managed within timelines.

The WAA tool is now the reference implementation used for pre-season planning by stakeholders. Future work is planned to evolve the application to support mid-season forecasts usable in a more ongoing, operational context.

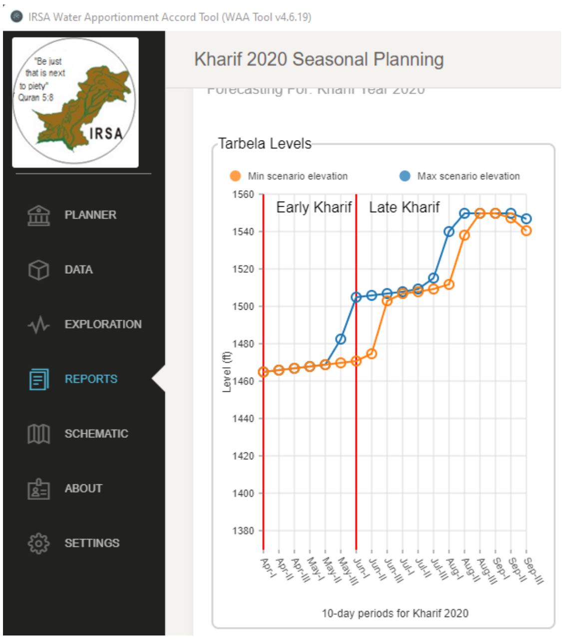

Figure 1. WAA Tool 4.6 desktop user interface

Keywords: *Reservoir operation, seasonal forecast, native software interoperability*

1. INTRODUCTION

The Indus Basin Irrigation System (IBIS) is the world's largest continuous irrigation system and supports food production, energy generation and stock, domestic and industrial supply for the nation of Pakistan. The agricultural sector accounted for 19.8% of the gross domestic product in 2015–2016 and was a source of livelihood for 42.3% of Pakistan's labor force (Anwar and Bhatti, 2018). Pakistan is already a water-stressed country, particularly vulnerable to the effects of climate change (Ahmad et al., 2021) and its growing population of 230 million persons may reach 300 million by 2050 (Kirby and Ahmad, 2022).

Conflictual and contested situations in the sharing of waters between populations in the provinces of Pakistan are dating back to a time well before the independence of the country in 1947 (Ahmed et al., 2018; Ahmed, 2019). The landmark 1991 Inter-Provincial Water Apportionment Accord (Government of Pakistan, 1991) is a consensus agreement where broad principles on how the water resource is to be allocated between provinces in the IBIS are described. However, how these principles are interpreted and implemented is a detailed, complex process that is understood by a few experts. The Water Apportionment Accord (WAA) Tool version (Ahmad et al., 2022) with pre-season water allocation planning capabilities is an output from work undertaken under an Australia-Pakistan Water Memorandum of Understanding. This work was undertaken by the CSIRO in close cooperation with federal and state water agencies in Pakistan.

The present paper gives an overview of the WAA Tool from a software design viewpoint, within the context of the project history. We cover the main design and technological decisions made to develop a product consistent with prior existing planning tools, suitable for use in the current IT environments, and evolvable in the future to address new demands for mid-season operational reservoir management. We only give a summary of the human, organisational and procedural context of the water sharing. More extensive and detailed information can be found in Ahmad et al. (2022).

2. BACKGROUND AND PRE-SEASONAL WATER-SHARING FORECASTS

Given the monsoonal regime of the annual climate cycle, pre-season planning of major reservoir operations for the Kharif (wet, April to September) and Rabi (dry, October to March) seasons is performed every September and March. Figure 2 outlines the steps in the sharing process. In step 1, lower and upper scenarios for inflows from major rivers are forecast, using the recent past system inflows as predictors and long-term streamflow records. In step 2, estimates of unaccounted system losses to be expected are made, guided by recent years losses. In step 3 and 4, reservoir releases at a time step of ~10 days from major reservoirs at Tarbela and Mangla are then iteratively determined for each scenario to match filling targets at set times, operating and safety constraints, equity of access and target outflows at the estuary of the Indus River. Reservoirs are operated in harmony when emptying, i.e. trying to keep a similar filling percentage. Step 5 balances resources shortages between two sub-areas of the basin, the Indus and Jhelum-Chenab systems, the latter topologically flowing into the former. In step 6 water share shortage is shared between provinces, and optionally shortages are optionally equalised over time (10-day periods) to avoid the risk of large shortages at critical cropping times. Step 7 is performed within provinces and out of scope for this paper.

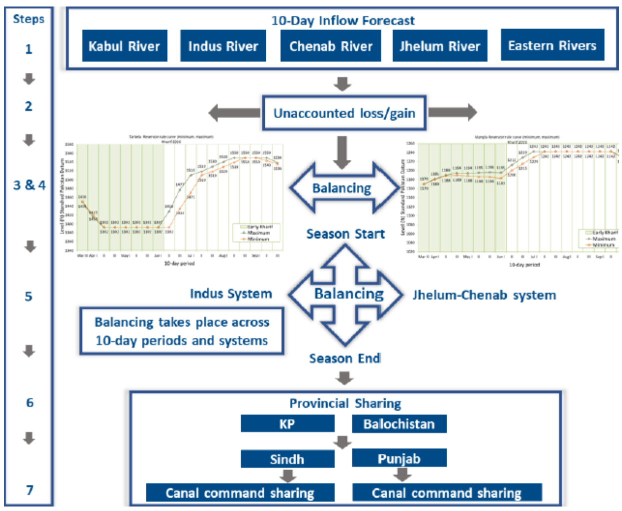

Figure 2. Main steps of the allocation process in the IBIS from (Ahmad et al., 2022)

Historically, this process has been done using Excel spreadsheets, exchanging data files and spreadsheets over transfer media such as emails. Each of the steps incurred some careful manual adjustment of spreadsheet inputs to achieve the multiple objectives of the step, testament to the intricate, expert knowledge of the system.

River system and/or economic models have previously been built to simulate the processes in the IBIS (Ahmad et al., 2018; Ahmed et al., 2018; Podger et al., 2021; Stewart et al., 2018; Yang et al., 2013). Some of them included a seasonal allocation procedure reflecting the Apportionment Agreement, but primarily addressed other biophysical or water management processes at a finer scale. As such, they were not designed solely to support the pre-seasonal planning process of stakeholders. There may have been unpublished attempts at building a dedicated software tool to aid the bi-annual pre-seasonal water forecast, but if so by 2019 none had been adopted in preference to the established procedure and practices.

3. SOFTWARE IMPLEMENTATION

3.1. Product requirements and reference implementations

Until 2019, the project elicited and iteratively captured the steps of the seasonal water-sharing method into software using the programming language Fortran 95. The key person in the project writing software was most productive with Fortran and a logical way to progress iteratively toward a proof-of-concept WAA tool, including a Windows-only desktop user interface. This Fortran implementation included novel iterative algorithms for each of the steps and matched the output forecast by the existing reference implementation in spreadsheets.

Several needs however were not addressed optimally by a tool coded in Fortran. Modern user interface toolsets are not accessible from it, including the rich ecosystem of Web-based user interface componentry. A partial or complete reimplementation was deemed necessary to access state of the art user interface. The long-term maintenance and evolution of a software application by modelers and software developers in Pakistan also demands a consideration of technology choice to allow a larger freedom to operate and autonomy. The water-sharing model may be conceptually complicated and intricate, but the runtime of a scenario remains small. A transition to other programming language was possible and preferable to be managed by a large pool of trained engineers and developers now and in the future. Another consideration for the design and technology choice is liberal licensing of third-party software and no or very minimal pricing encumbrance.

Lastly, we envisage a future version as a web application accessible from a browser, accessing to a centrally curated and up to date database informing the water-sharing process, rather than a desktop application. However, a desktop product remains the only feasible and practical option in the immediate future, closer to the current form of input data and organisational settings.

Given these requirements and contextual settings, we summarise in the next section the status in February 2023 of the software stack used for the re-implemented WAA Tool.

3.2. Product architecture and technologies

Python is a very popular programming language in science and engineering, with some capabilities syntactically similar to Fortran for numerical computing. In particular numpy (https://www.numpy.org) is a natural equivalent for multi-dimensional array indexing. *xarray* (Hoyer et al., 2023) uses named dimensions on top of numpy. It is typically used to handle large data, but is beneficial for legibility when handling the smaller multidimensional arrays in the WAA Tool. The core model logic of the tool takes the form of a Python package *waa* ("WAA model" in Figure 3). This package uses type hints to enhance maintainability and is documented with "docstrings" that are used to produce a searchable HTML reference documentation.

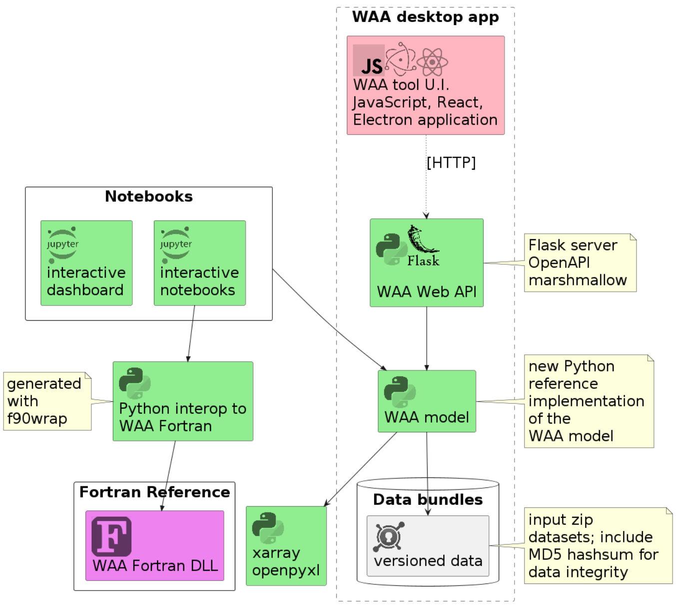

Figure 3. The salient software and data components of the latest WAA Tool desktop application, and the assets supporting the seamless interactive validation and testing against the proof-of -concept implementation in Fortran.

Producing reports that contain the right information laid out in a specific format is important for the decision process, and the archiving of these documents. To fit in the existing practices using spreadsheets, results of the tool can be exported to a report by populating programmatically Excel spreadsheet templates. This is possible using the open source third party library openpyxl (https://openpyxl.readthedocs.io).

Given that there is a possibility to move in the future to a web browser interface, the JavaScript library ReactJS (https://reactjs.org) and React Boostrap (https://react-bootstrap.github.io) were chosen to build the user interface. This is in preference to one of the Python libraries for desktop or web user interface, as this is not the strongest part in the Python ecosystem. The Electron framework (https://www.electronjs.org) is used to package this “web” front end as a desktop interface, as this is identified as the suitable access arrangement for users for the time being. A python package “waasrv”, separate from the model logic, hosts a server with a Web API using Flask (https://palletsprojects.com/p/flask), accessible by the user interface component. To adhere to the principles of a RESTful API, the client user interface is designed upfront to store the state of the application (parameters, inputs, options), rather than overly relying on the server to be stateful, to an extent practically feasible. To ease state management in the client, the state container Redux (https://react-redux.js.org) is used in combination with React, which is a typical usage pattern. Redux saga (https://redux-saga.js.org) is used to manage asynchronous operations such as HTTP requests, executing I/O operations.

Given the reliance on exchange of datasets between agencies in preparation to the planning process, there is a significant risk of discrepancies. A centralised database with versioning capabilities requires inter-organisational data arrangements that may be premature. Instead, we opted for a pragmatic and “low-tech” option to promote data integrity and versioning. We use date stamps and an MD5 hash sum in the name of a zipped file to prevent input data discrepancies between stakeholders involved in the decision process.

Figure 3 includes components relating to the Fortran implementation. These are now extraneous to the current WAA Tool. It would have been convenient to gloss over it in this paper, but they played an important role transitioning a trusted Fortran implementation to an implementation in pure Python. At a higher level it would be an interesting exercise to reflect on the project history and design decisions over its lifetime using the framework presented by Hamilton et al., (2022). The next section will focus on an overview of the technical setup and may be helpful to readers porting legacy Fortran models.

3.3. Porting from Fortran to Python

The water-sharing procedure captured in the reference implementation in Fortran may have a small runtime, but it is nevertheless complicated. Porting it to Python in a limited timeframe was a critical and risky step that needed to be managed with intermediary testing steps to end up with correct, identical results. A re-implementation from first principles is often preferable from a design standpoint, but in this case unlikely to succeed numerically, given that at that point the Fortran was more a point of truth than written documents. A possibility to fall back on the Fortran implementation of the model was also necessary as a precaution. The present section covers some key technical aspects in the hope this helps a reader facing a similar situation.

The approach taken to manage the transition is in spirit like that described in Freebairn et al. (2005), prying open the reference implementation for intermediary model states and compare side by side for discrepancies at a finer granularity. The accepted WAA Tool in Fortran is a monolithic executable program, launching as a user interface or as a command line tool with parameters. It outputs results to a text-based ASCII report with tables formatted for human eyes but difficult to ingest programmatically for testing. Some intermediary data and log information are also present, but again costly to produce for a variety of cases and ingest for testing.

Jupyter computational notebooks are one working environment to iteratively explore data and code. An ideal testing environment in our case is to be able to access directly the Fortran code from a Python notebook, and invoke subroutines to get intermediary states as a basis for unit testing. Aside from the reengineering to make the Fortran code compilable as a shared library, feasibility was depending on the availability of “glue code” generator between Fortran 90 and Python. As of October 2019, the venerable f2py (https://numpy.org/doc/stable/f2py) surprisingly appeared to still be supporting only Fortran 77 and was not suitable to generate wrappers for the Fortran 90 code of the WAA tool. However, by that date Xiong et al. (2020) had released f90wrap to generate Python wrappers for Fortran 90/95 code, which proved suitable to generate the substantial amount of wrapper code we needed.

Having access to a finer granularity of software component allowed for an easier verification of each modelling step, and the interactive construction of data sets for unit tests. Given the intricacies of the codebase and some of the arcane nature of historical methodological legacies, it would have been difficult to manage the port solely with Fortran program outputs as the sole reference results available.

The complicated nature of the algorithms implemented to achieve water sharing under constraints is typically reflected in code that is used to capture knowledge iteratively. This was the case for the Fortran codebase, and the port to Python kept some of the code structure to maintain comparability. Once a set of unit tests were established, re-factoring the code with a safety net was possible. The Python package radon (https://radon.readthedocs.io) is one way to have static code metrics, most notably cyclomatic complexity, to identify priority areas to reduce this complexity and facilitate readability and maintenance.

4. APPLICATIONS, IMPACT AND POSSIBILITIES

Several seasons were planned since 2018 using versions of the WAA Tool, in parallel with the established process and Excel spreadsheet, to validate the numerical behaviour and refine features needed for the user experience and capabilities. The WAA tool at version 4.6.9 was adopted in September 2022 as the reference decision support software for pre-seasonal planning of the Rabi 2022-23 dry season, superseding the decades-old manual process (Khaleeq Kiani, 2022).

Additional options have now been added in the WAA Tool compared to the Fortran implementation. One is the option to share water shortages among more provinces than the biggest two (Punjab and Sindh), as was historically the case. An option to re-adjust maximum and minimum scenarios to prevent curves crossing over, which can happen in some circumstances, and is correct and innocuous, but distracting from decisions.

A more important recent addition (September 2021) is the capability to adjust water sharing so that the seasonal totals for each province match their expected fair share. One of the side effects of the complexities of water sharing over each 10-day periods and over time was that sharing fairly over each 10-day period usually does not lead to the expected seasonal total share. This discrepancy, however small, has historically been a trigger of tension between the provinces sharing water shortages. A novel algorithm was devised to rebalance seasonal total shared shortages robustly and reliably to the expected values, while minimising departure from initial provincial water shares for each 10-day period of the season. The algorithm is explained mathematically and graphically in detail in Ahmad et al. (2022).

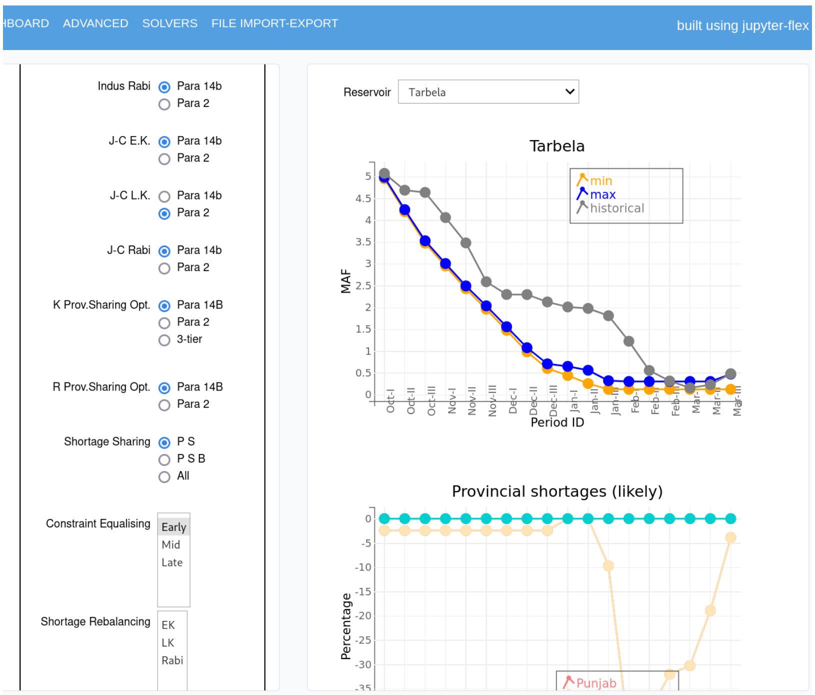

Figure 4. Prototyping a custom, interactive test dashboard written in Python using ipywidgets, voilà and jupyter-flex

Further improvements are now requested for the tool. For instance, while the user interface is fully interactive and the overall runtime enables dynamic output refresh, it was not designed as a customisable technical dashboard. Porting the model to Python brings the benefit of a large and evolving ecosystem. To complement the "terse" unit testing we have written in Python a web dashboard (Figure 4). This does not feature the level of polish of *ReactJS* components but is a lower cost option to explore alternate interface designs to make the model more transparent to stakeholders.

Having the model implemented as a Python package with programmatically accessible functions also facilitates, at the very least logistically, further scientific and impact studies. Given the vulnerability of Pakistan to the impacts of climate change, assessing the effects on the IBIS of predicted shifts in the quantity, timing and variability of inflows will be much simpler to set up in a jupyter notebook, with minimal impedance from data wrangling and format conversion. A second possible line of investigation is to assess the use of novel seasonal forecasting methods for the inflows into the IBIS, for instance using streamflow forecasting methods evaluated in Charles et al. (2018).

At the time of writing, further engagement and development work is explored building upon the existing software stack. It is important to undertake these with actions such that stakeholders in Pakistan have the freedom to operate with these software assets. One possible project is the elicitation and documentation of practices and processes undertaken within the wet or dry season, if forecasts need to be revised.

Work has also commenced on another major dam on the Indus River, the Diamer Basha dam (https://en.wikipedia.org/wiki/Diamer-Bhasha_Dam). This will require within a few years a significant reassessment of the mechanisms for water sharing in the IBIS and naturally entail changes to the WAA Tool.

5. CONCLUSION

The WAA tool is a contemporary desktop application to manage the pre-seasonal planning of large-scale water sharing and major reservoir operations in the Indus Basin Irrigation System (IBIS) in Pakistan. From September 2022 onwards it is adopted as the reference application for pre-seasonal planning for the IBIS.

After an initial phase writing a proof-of-concept implementation in Fortran, the application is now implemented with the programming languages Python and JavaScript to ensure there is a large pool of developers in Pakistan skilled for the maintenance and evolution of the software assets. The user interface is currently accessible as a desktop application but designed from the ground up with Web technologies, as we foresee a possible evolution to a Web application along with organisational change for water data management.

Implementing the water-sharing model engine in Python gives access to a large ecosystem of science and engineering capabilities. We used f90wrap to interoperate with the initial proof-of-concept Fortran code and sustainably port the code to Python while setting up regression tests and unit tests. The package openpyxl enables the production of model reports as Microsoft Excel files fitting in the existing business processes. The Python implementation of the WAA Tool now in use includes a novel algorithm to rebalance seasonal shortages amongst provinces, sealing the end of the transitional reliance on the proof-of-concept Fortran implementation.

By moving to more contemporary software technologies, future modelling endeavours such as the integrated assessments of novel streamflow forecasting methods or climate change impacts are now logistically easier to undertake.

ACKNOWLEDGEMENTS

We thank the Indus River System Authority (IRSA), Water and Power Development Authority (WAPDA), Sindh Irrigation Department (SID) and Punjab Irrigation Department (PID) for their provision of data, technical advice and review, and the Khyber Pakhtunkhwa Irrigation Department (KPID) and Balochistan Irrigation Department (BID). Their trust and sustained engagement over years towards this outcome is gratefully acknowledged.

The WAA Tool is an output of ongoing work started in the Australian Department of Foreign Affairs and Trade (DFAT) Sustainable Development Investment Portfolio (SDIP) Phase II (2016–20) project in the Indus basin Pakistan. The prototype was developed into the WAA Tool in 2020 with financial support from the Government of Australia through its Australian High Commission Islamabad and CSIRO. The current improvements and documentation are supported through a research and development activity (WAC/2021/103) by the Australian Centre for International Agricultural Research (ACIAR).

REFERENCES

Ahmad, M., Stewart, J.P., Podger, G.M., 2018. Modelling the surface water allocations and deliveries in the world's largest continuous irrigation system, Indus Basin Pakistan, Geophysical Research Abstracts.

Ahmad, M. ud D., Peña-Arancibia, J.L., Yu, Y., Stewart, J.P., Podger, G.M., Kirby, J.M., 2021. Climate change and reservoir sedimentation implications for irrigated agriculture in the Indus Basin Irrigation System in Pakistan. J Hydrol (Amst) 603, 126967. https://doi.org/10.1016/J.JHYDROL.2021.126967

Ahmad, M., Yu, Y., Cuddy, S., Perraud, J.-M., Podger, G., Freebairn, A., Seaton, S., Shah, S.M.M.A., Rana, M.K.I., Khero, Z., Bodla, H.U., Farooq, M., Khan, M.A., Khan, T.A., 2022. Bringing transparency and consistency to Pakistan's seasonal water planning decisions: 1991 inter-provincial Water Apportionment Accord (WAA)Tool User Guide and Reference Manual. Canberra.

Ahmed, W., Rais, M.N., Bano, R., Tamaddun, K., Ahmad, S., 2018. Water Sharing, Governance, and Management among the Provinces in Pakistan Using Evidence-Based Decision Support System, in: World Environmental and Water Resources Congress 2018: Watershed Management, Irrigation and Drainage, and Water Resources Planning and Management - Selected Papers from the World Environmental and Water Resources Congress 2018. American Society of Civil Engineers (ASCE), pp. 220–233. https://doi.org/10.1061/9780784481400.020

Ahmed, Z., 2019. Pakistan's Water Crisis and Indus River System: Revisiting National Security. Journal of Political Studies 26, 171.

Anwar, A.A., Bhatti, M.T., 2018. Pakistan's Water Apportionment Accord of 1991: 25 Years and Beyond. J Water Resour Plan Manag 144, 05017015. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000831

Charles, S.P., Wang, Q.J., Ahmad, M.U.D., Hashmi, D., Schepen, A., Podger, G., Robertson, D.E., 2018. Seasonal streamflow forecasting in the upper Indus Basin of Pakistan: An assessment of methods. Hydrol Earth Syst Sci 22, 3533–3549. https://doi.org/10.5194/HESS-22-3533-2018

Freebairn, A., Rahman, J., Seaton, S., Perraud, J.-M., Hairsine, P., Hotham, H., 2005. Development of an automated testing tool for identifying discrepancies between model implementations, in: MODSIM05 - International Congress on Modelling and Simulation: Advances and Applications for Management and Decision Making, Proceedings.

Government of Pakistan, 1991. Water Apportionment Accord: Apportionment of the Waters of Indus River System between the provinces [WWW Document]. URL http://pakirsa.gov.pk/WAA.aspx (accessed 2.19.22).

Hamilton, S.H., Pollino, C.A., Stratford, D.S., Fu, B., Jakeman, A.J., 2022. Fit-for-purpose environmental modeling: Targeting the intersection of usability, reliability and feasibility. Environmental Modelling & Software 148, 105278. https://doi.org/10.1016/J.ENVSOFT.2021.105278

Hoyer, S., Roos, M., Joseph, H., Magin, J., Cherian, D., Fitzgerald, C., Hauser, M., Fujii, K., Maussion, F., Imperiale, G., Clark, S., Kleeman, A., Nicholas, T., Kluyver, T., Westling, J., Munroe, J., Amici, A., Barghini, A., Banihirwe, A., Bell, R., Hatfield-Dodds, Z., Abernathey, R., Bovy, B., Omotani, J., Mühlbauer, K., Roszko, M.K., Wolfram, P.J., 2023. xarray. https://doi.org/10.5281/ZENODO.7616056

Khaleeq Kiani, 2022. Govt to adopt new water data tool, ditches old manual system. Dawn.

Kirby, M., Ahmad, M. ud D., 2022. Can Pakistan achieve sustainable water security? Climate change, population growth and development impacts to 2100. Sustain Sci 17, 2049–2062. https://doi.org/10.1007/S11625-022-01115-0/TABLES/4

Podger, G.M., Ahmad, M.-D., Yu, Y., Stewart, J.P., Shah, S.M.M.A., Khero, Z.I., 2021. Development of the Indus River System Model to Evaluate Reservoir Sedimentation Impacts on Water Security in Pakistan. Water (Basel) 13, 895. https://doi.org/10.3390/w13070895

Stewart, J., Podger, G., Ahmad, M., Shah, S.M.A., Bodla, H., Khero, Z., Rana, M.K.I., 2018. Indus River System Model (IRSM) - a planning tool to explore water management options in Pakistan: model conceptualisation, onfiguration and calibration. https://doi.org/10.25919/5B7326BCD1554

Xiong, Q.Y., Huang, S.Y., Yuan, Z.G., Kermode, J.R., 2020. f90wrap: an automated tool for constructing deep Python interfaces to modern Fortran codes. Journal of Physics: Condensed Matter 32, 305901. https://doi.org/10.1088/1361-648X/AB82D2

Yang, Y.-C.E., Brown, C.M., Yu, W.H., Savitsky, A., 2013. An introduction to the IBMR, a hydro-economic model for climate change impact assessment in Pakistan's Indus River basin. Water Int 38, 632–650. https://doi.org/10.1080/02508060.2013.830691