Response to Reviewer RC1

Title: Two-dimensional Differential-form of Distributed Xinanjiang Model

Authors: Jianfei Zhao, Zhongmin Liang*, Vijay P. Singh, Taiyi Wen, Yiming Hu, Binquan Li, Jun Wang

Manuscript ID: hess-2024-377

We would like to sincerely thank you for thoroughly reviewing our manuscript. All your comments have been carefully addressed, and a point-by-point response is provided below.

For better readability, the point-by-point response is formatted as follows:

- Reviewer's comments are shown in black
- Authors' responses are shown in blue
- Revisions to be incorporated in the revised manuscript are highlighted in red
- Figures used only in this response will have the prefix "R", such as Figure R1
- Figures without prefix "R" are numbered as they appear in the manuscript

Overall comments:

This work is a typical representation of the forward development in the current hydrological modeling field. By leveraging existing advanced numerical techniques, it reconstructs the traditional empirical, conceptual, and lumped hydrological model, transforming it into modern distributed hydrological model that can represent spatial heterogeneity and are based on complete systems of differential equations. Overall, I think this work is of great practical significance. In addition, the author's writing is also quite mature and concise. Nevertheless, I still believe that this work should undergo major revision before it is suitable for publication, as the presentation of the results section remains insufficient for a mature study.

Thank you for your positive remarks on our work, particularly for acknowledging and highlighting our efforts. Your comments are invaluable, and we have carefully considered each point to enhance the clarity and maturity of the subsequent revised manuscript. We hope that these changes could effectively address your concerns.

Major concerns:

1. I have reviewed the model code provided by the author and noticed that it is relatively concise pure Python code and does not include many additional dependencies. I believe this is a strong advantage, as it suggests broader usage scenarios and greater potential for expansion. However, as an interpreted language, Python has a typical limitation in comparison to compiled languages like C, namely its lower efficiency. As a potential application for large domain modeling, I am concerned about the computational

efficiency of the TDD-XAJ model. The author should at least provide a reference for efficiency, and clarify whether transitioning from a 1D modeling framework to a 2D framework would result in a significant decrease in computational efficiency. I believe the author should add a section to address these issues.

In addition, please include a description of the programming environment of the model, as well as the dependencies used, to help users clearly understand the usage scenarios.

Thank you for taking the time and effort to review the code we provided. The model was implemented solely in Python, and to ensure readability, understandability, and ease of application and potential future extensions, we used only essential libraries, keeping the code concise while maintaining completeness. We are grateful for your recognition of this manner and your positive feedback.

We understand your concern regarding the computational efficiency of the TDD-XAJ model. Just as you pointed out, Python is an interpreted language, does lag behind compiled languages such as C or Fortran in terms of execution speed. To address this, we will add a section in Result and Discussion part of the revised manuscript, which is shown below:

"**4.4 Computational efficiency of the model**

The implementation of the TDD-XAJ model is detailed in Appendix A. The Python language was chosen for its robust ecosystem in hydrology and earth system sciences (Stacke and Hagemann, 2021; Murphy et al., 2024), which facilitates rapid development and experimentation. While Python, as an interpreted language, is generally less computationally efficient than compiled languages, it benefits from libraries like NumPy, which rely on compiled languages for underlying computations. To enhance computational performance, we leveraged NumPy's vectorized operations and optimized functions, which are generally faster than native Python code. However, we recognize that NumPy may still be slower in certain cases, such as element-wise calculations, compared to compiled languages. This is a well-known limitation within the Python community, and solutions like Just-In-Time (JIT) compilation have been proposed (Lam et al., 2015), which convert frequently executed script code into machine code with further automatic optimizations. Although this manuscript primarily focuses on presenting the theoretical aspects of the TDD-XAJ model, we plan to optimize the code, including exploring parallelization, in future work."

Transitioning from a 1D modeling framework to a 2D framework does not result in a significant decrease in computational efficiency. To clarify this, we repeated the models used in the slope concentration methods comparison experiment and recorded the runtime. The details and analysis will be added to the new Section 4.4 of the revised manuscript, which is provided below:

“To compare the computational efficiency of the 1D and 2D slope runoff concentration methods, we repeated the models used in corresponding comparison experiment for 10 times, and recorded the runtime for each. The average runtime values, obtained using an AMD 5950X CPU (3.4 GHz) in single-core mode, are summarized in Table 5.

Table 5. Average runtime recorded in the slope runoff concentration methods comparison numerical experiment.

Surface comparison scenario (s)			Subsurface comparison scenario (s)		
Method	Single-slope case	Double-slope case	Method	Single-slope case	Double-slope case
1D diffusion wave	28.41	28.32	1D linear reservoir	2.10	2.06
2D diffusion wave	29.02	29.80	2D linear reservoir	1.13	1.12

As shown in Table 5, replacing 1D slope runoff concentration methods with 2D methods incurs minimal computational overhead while potentially enhancing efficiency. For diffusion wave method used for surface runoff concentration, the runtime for the 2D form increases 2.1% and 5.2% compared to the 1D form in the single-slope and double-slope case, respectively. Conversely, for linear reservoir method applied to subsurface runoff concentration, switching from 1D to 2D form reduces runtime by at least 45.6% across both test cases. The numerical computation procedure for the slope runoff concentration methods mainly involves two steps: flux calculation and state variable update. In flux calculation step, the 1D and 2D diffusion wave method require calculating once per cell (based on GIS-derived flow direction) or twice (in both x and y directions), respectively. For linear reservoir method, both the 1D and 2D forms require calculation only once per cell, and the outflow flux evaluated is further decomposed into x- and y- directional components based on slope aspect for the 2D form. When updating state variables, both the diffusion wave and linear reservoir methods in their 1D form require logical judgments based on GIS-derived flow direction to identify the inflow cells. In their 2D form, this can be directly determined using cell indexing. The element-wise logical judgment is relatively time-consuming and is comparable to a one-time flux calculation per cell. Therefore, for the diffusion wave method, transitioning from 1D to 2D form does not result in a significant increase in runtime, but for the linear reservoir method, the computational overhead is effectively reduced.”

As suggested, an appendix will be included in the revised manuscript detailing the programming environment and library dependencies used in the model, which is shown below:

“**Appendix A: Implementation details of the TDD-XAJ model**

The TDD-XAJ model is implemented in Python 3.9, taking advantage of its extensive ecosystem, as well as cross-platform capabilities. To maintain readability and facilitate understanding, only essential libraries are used, ensuring the model is both accessible and easily extendable in the future. A list of these libraries is provided in Table A1. NumPy plays a crucial role, providing vectorized operations and optimized computational functions that enhance performance compared to native Python. Other libraries serve supporting roles, such as model configuration and handling file input/output.

Table A1. Libraries utilized in the implementation of the TDD-XAJ Model.

Category	Name	Version	Description
Core	NumPy	1.26.4	A basic scientific computing library for efficient numerical operations
Auxiliary	PyYAML	6.0.1	Library for reading YAML files used for model configuration and settings
	Pandas	2.2.1	A high-level library for data manipulation and analysis
	OpenPyxl	3.1.5	Backend support for reading and writing Excel files in Pandas
	treelib	1.7.0	Efficient implementation of tree data structure

"

2. I have some confusion regarding the channel unit modeling in Figure 4. In the previous sections (Section 2.1 and Figure 1), the channel units are explicitly defined as segments, with their lengths clearly specified in three ways. In these cases, the width of the channel units is significantly smaller than the grid size so that the rainfall can be neglected. However, in this test case (Figure. 4), the grid size (5m x 5m) is significantly larger than the channel width (20m). Does this mean that the channel units need to encompass multiple grids? I would ask the author to provide a corresponding diagram to clarify how spatial discretization and channel unit modeling are handled in the test cases.

A similar issue also applies to real basin modeling. Could you provide a diagram to illustrate how spatial discretization is carried out in the Tunxi watershed, including the definition of grid units and channel units?

Thanks for your comments. The TDD-XAJ model uses DEM grid cells as its fundamental computational units, which means the simulation domain is spatially discretized with square structure grid (see Figure R1 below). These cells are explicitly categorized into two types: (1) channel cells, which containing river channels, where both slope concentration and channel concentration processes are considered, including water exchange between the slope and the channel, and (2) non-channel cells, which do not contain river channels and only slope concentration process is considered. In channel cells, the channel concentration process is governed by one-dimensional diffusion wave equations. This allows the channel to be conceptualized as a line feature within the model, with its width treated as a cross-sectional attribute of the one-dimensional equations. The spatial relationship between the channel line feature and channel cells adopts two topologies: intersecting (channel passes through cell interiors) and adjacent (channel aligns with cell boundaries). In real-world watershed applications, the channel width is generally smaller than the grid size. Thus, we implement the intersecting topology by positioning the channel line through cell centers (Figure R1a). In the synthetic V-shaped watershed test case, where the channel width (20m) is significantly larger than the grid resolution (5m), we adopt the adjacent topology to keep physical realism and avoid representing the channel with multiple cells (Figure R1b).

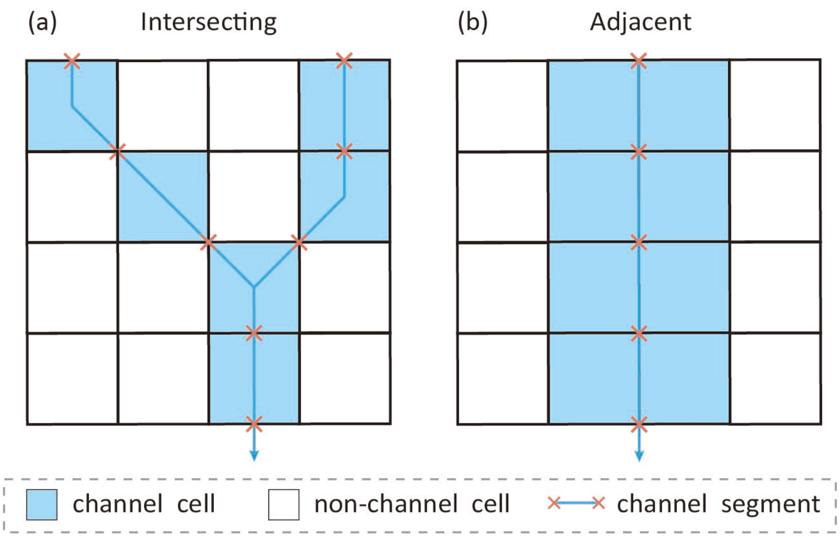

Figure R1. Diagram of spatial discretization and the arrangement of channel line features (with two spatial topological relationships between the channels and slopes).

Figure 5 in our manuscript initially presented the location of the Tunxi watershed and its gauging stations. To illustrate the spatial discretization in the Tunxi watershed, including the definition of grid (non-channel) units and channel units, we have made further extensions based on this (see Figure 5b) and the revised Figure 5 is shown below:

" Line 421:

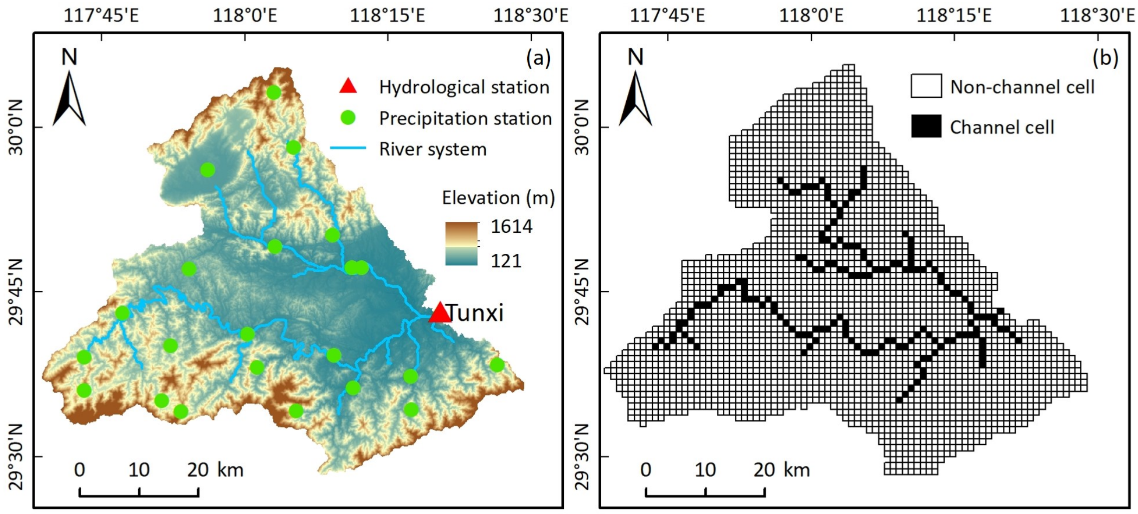

Figure 5. Location and gauging station distribution of the Tunxi watershed (a), and the spatial discretization of the watershed, including channel and non-channel cells (b). "

3. Since the author emphasizes the advantages of the 2D method in capturing microtopography and

presents it as a typical case of distributed modeling, the author should at least provide some results for spatial simulations. This would better illustrate the model's advantages and provide supporting evidence for the attribution.

Specifically, the author could provide some degree of spatial validation (for example), which is highly valued in the hydrological modeling community. If this proves challenging (due to data limitations or other reasons), an alternative could be to include a test case with more varied slopes (such as two or three different slope changes in the y-direction). In this more complex test case, testing the consistency between slope variations and simulation results would better highlight the advantages of 2D modeling.

Thanks for your valuable comments. To address this, we have further included the spatial simulation results in addition to the simulated hydrograph in 2008 (Figure 9 in manuscript, it’s revised version is shown below). We focused on the precipitation events from June 6th, 7th, and 8th, 2008 (just before the flood peak, with distinctive spatial precipitation distribution). On June 6th, there was almost no precipitation (Figure 9a), on June 7th, the precipitation was concentrated in the upper-left corner of the watershed (Figure 9b), and on June 8th, it was concentrated in the lower-right corner (Figure 9c). We provide the spatial simulation results for tension and free water storage on the three days. As shown in Figure 9d and 9g, the tension water storage of the watershed as a whole was not yet fully saturated, and the free water storage was almost zero. After June 7th, the tension water storage in the upper-left part of the watershed became saturated, and the rest of the watershed also became deeper due to precipitation (Figure 9e). The overall free water storage was also increased, with more significant increases in areas where precipitation was heavier, and vice versa (Figure 9h). On June 8th, the tension water in the lower-right corner of the watershed became fully saturated due to the coverage of precipitation in corresponding area (Figure 9f). For free water (Figure 9i), the storage in the lower-right part of the watershed also approached saturation, while the saturated areas in the upper part of the watershed quickly dissipated due to the faster recession of free water.

“ Line 566:

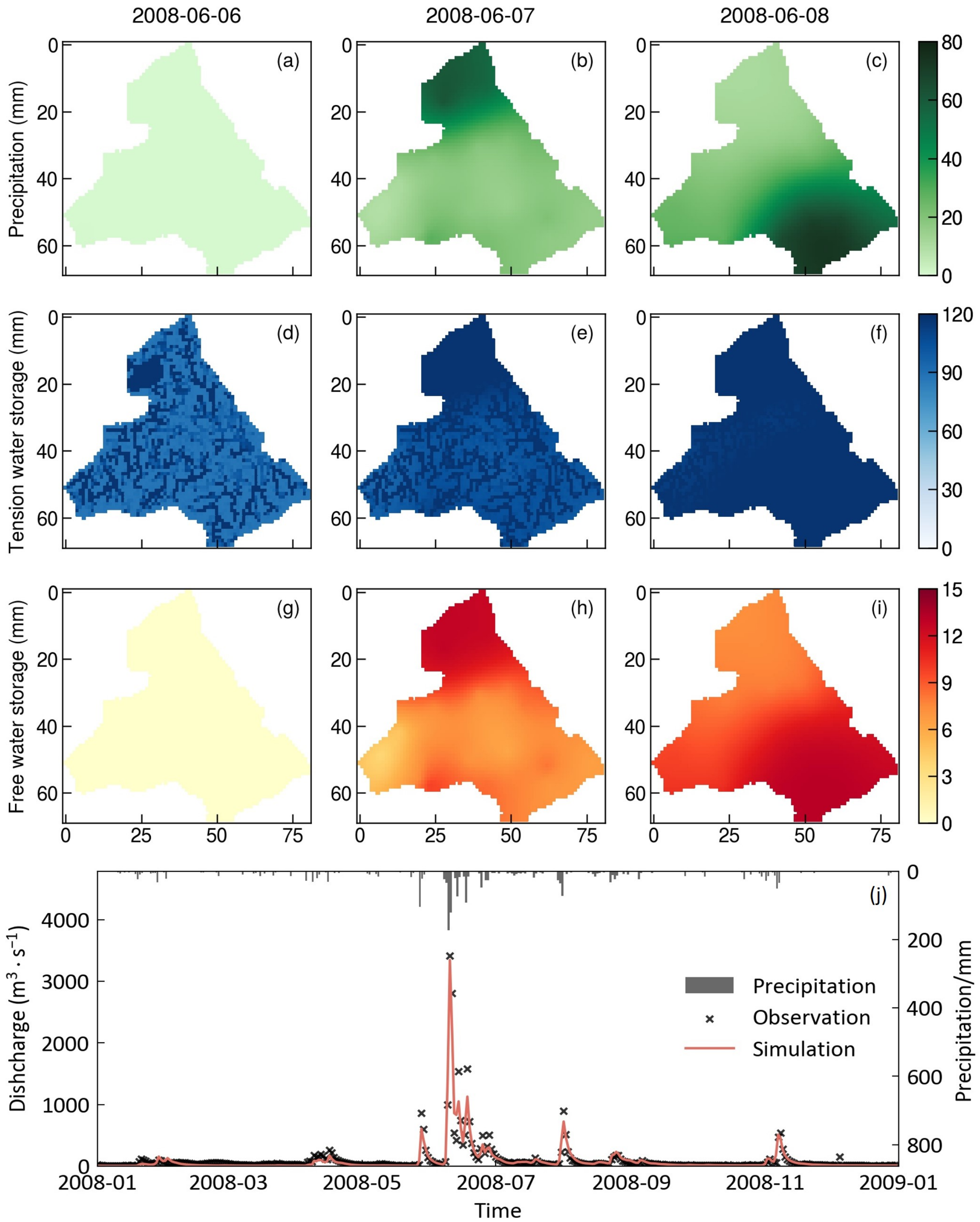

Figure 9. The spatial distribution of precipitation for three days in 2008 (a-c), along with the spatial distribution of tension (d-f) and free water storage (g-i) simulated by TDD-XAJ model for the three days, and the simulated hydrograph of the Tunxi watershed in 2008 using the model (j).”

Due to data limitations, we regret that we are unable to provide further spatial validation. As an alternative, and based on your suggestion, we have expanded the range of slope in the *y*-direction (S_{oy}). The original used S_{oy} of 0.00 (single-slope case) and 0.02 (double-slope case) have been extended to 0.00, 0.01, 0.02, 0.03, 0.04, and 0.05, while the slope in the *x*-direction (S_{ox}) is fixed to 0.05. For each case, both 1D and 2D runoff concentration methods are applied. The 1D methods use GIS-derived flow directions, and the comparison between the calculated and real flow directions is shown in Figure R2, and they are identical only when S_{oy} equals 0.00 and 0.05.

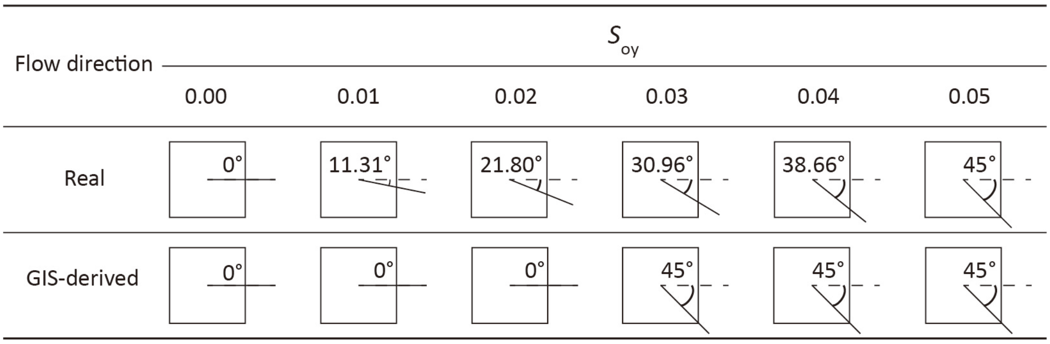

Figure R2. The comparison of real and GIS-derived flow direction under different *y*-directional slope (S_{oy}).

For surface runoff concentration, the 1D and 2D forms of diffusion wave (DW) method are used. Figure R3 and R4 illustrate the comparison of hillslope and channel outflow hydrograph for different S_{oy}, respectively. In the case of subsurface runoff concentration, the 1D and 2D forms of linear reservoir method (LR) are implemented. The hillslope outflow hydrograph is demonstrated in Figure R5, with channel outflow hydrograph further detailed in Figure R6. We used mean absolute error (MAE) to assess the difference between hydrographs simulated by the 1D and 2D forms of runoff concentration methods. From these figures, we can see that when the GIS-derived flow direction aligns with the actual flow direction (S_{oy}=0.00, and S_{oy}=0.05), the results of 1D form methods match those of the 2D form methods (see Figure R3a and R3f, Figure R4a and R4f, Figure R5a and R5f, Figure R6a and R6f). However, when there are discrepancies in derived and real flow directions, certain components in 1D form methods are neglected and its results show deviations.

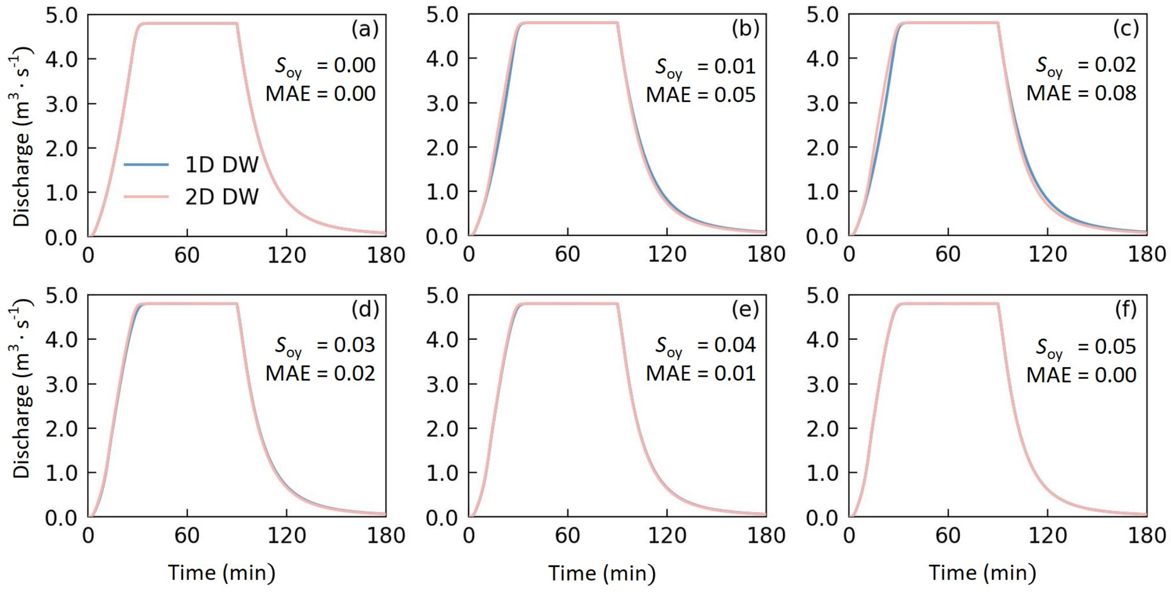

Figure R3. The hillslope outflow hydrograph of synthetic V-catchment test case with varying *y*-directional slope (S_{oy}) in surface comparison scenario. The one-dimensional (1D) and two-dimensional (2D) form diffusion wave methods (DW) were evaluated in this scenario. The difference between the outflow of 1D and 2D form methods was assessed with mean absolute error (MAE).

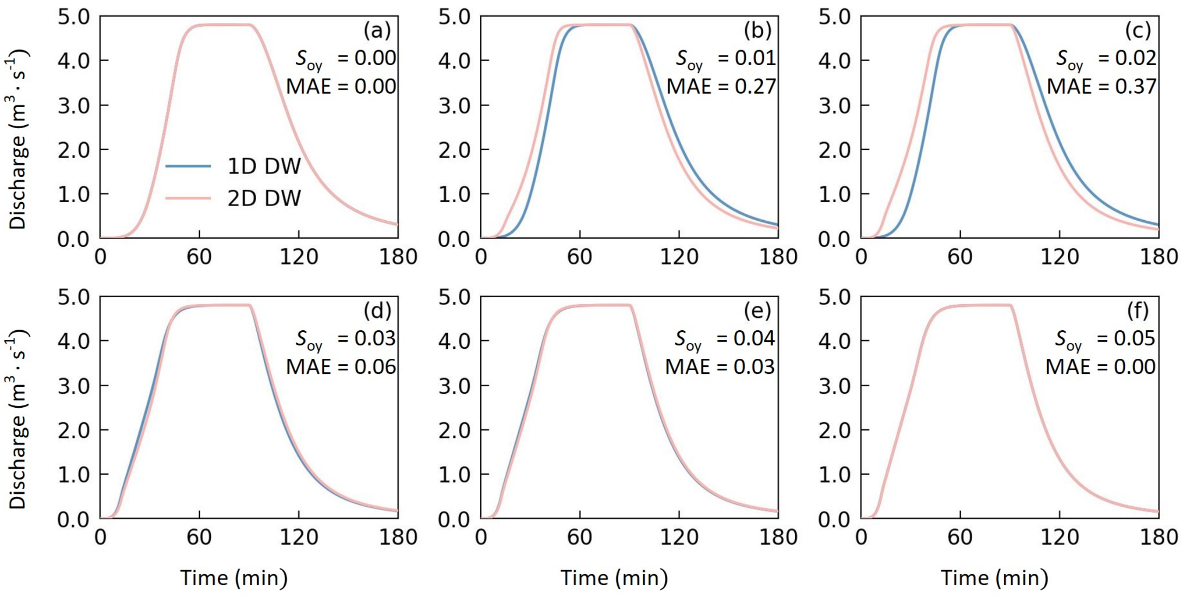

Figure R4. The channel outflow hydrograph of synthetic V-catchment test case with varying *y*-directional slope (S_{oy}) in surface comparison scenario. The one-dimensional (1D) and two-dimensional (2D) form diffusion wave methods (DW) were evaluated in this scenario. The difference between the outflow of 1D and 2D form methods was assessed with mean absolute error (MAE).

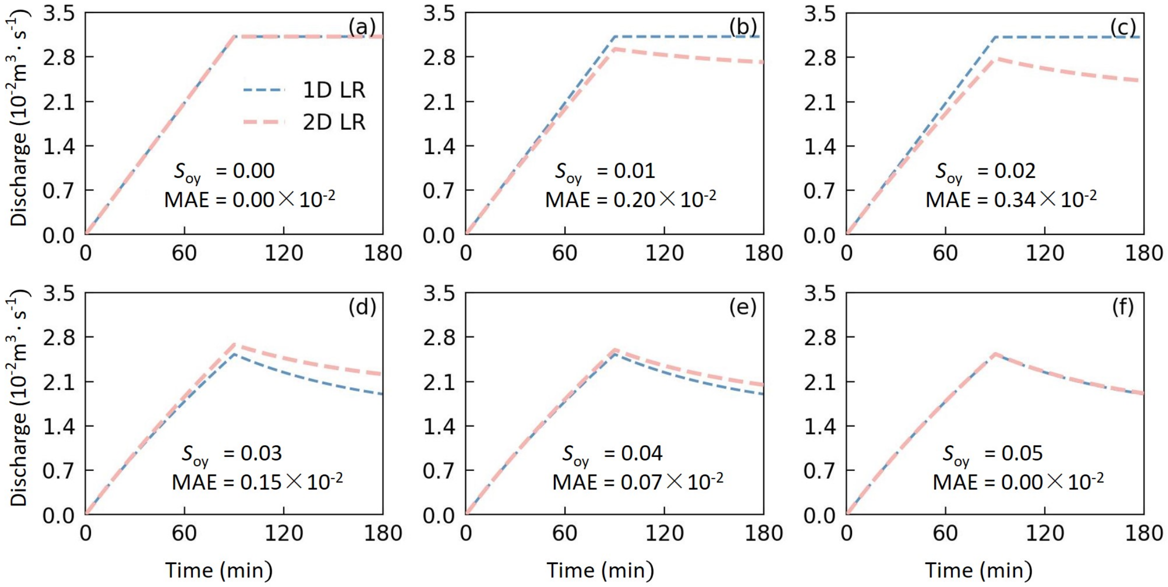

Figure R5. The hillslope outflow hydrograph of synthetic V-catchment test case with varying *y*-directional slope (S_{oy}) in subsurface comparison scenario. The one-dimensional (1D) and two-dimensional (2D) form linear reservoir methods (LR) were evaluated in this scenario. The difference between the outflow of 1D and 2D form methods was assessed with mean absolute error (MAE).

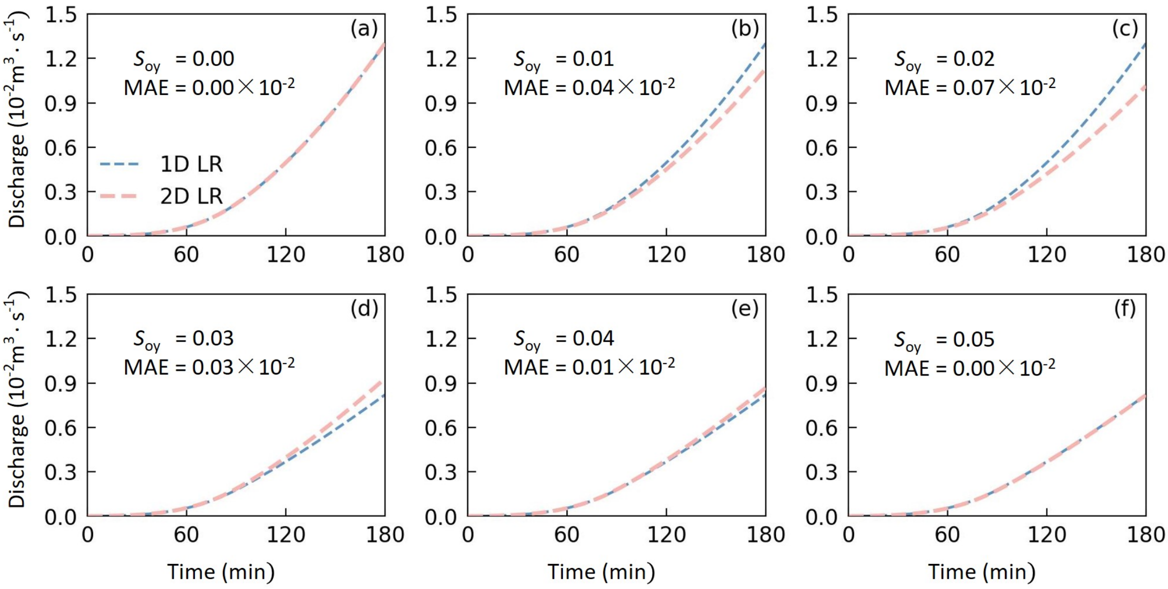

Figure R6. The channel outflow hydrograph of synthetic V-catchment test case with varying *y*-directional slope (S_{oy}) in subsurface comparison scenario. The one-dimensional (1D) and two-dimensional (2D) form linear reservoir methods (LR) were evaluated in this scenario. The difference between the outflow of 1D and 2D form methods was assessed with mean absolute error (MAE).

4. The differences between Figure. 7b and Figure. 7c suggest that 1D modeling overlooks the interflow component in the y-direction, although this is not particularly clear. Could you include one or two time profile plots corresponding to different stages in Figure. 6g to further support this conclusion?

Thanks for your valuable feedback. In Figure 7, we illustrate the differences between the 1D and 2D forms of the linear reservoir method for subsurface runoff concentration through Figure 7b and Figure 7d. As you pointed out, the contrast between these two sub-figures is not very clear. To address this, we have plotted the distribution of interflow storage completely and then zoomed in on the areas where differences are more noticeable. This visualization approach helps to highlight how the 1D model overlooks the interflow component in the *y*-direction. The updated Figure 7 is shown below:

"Line 452:

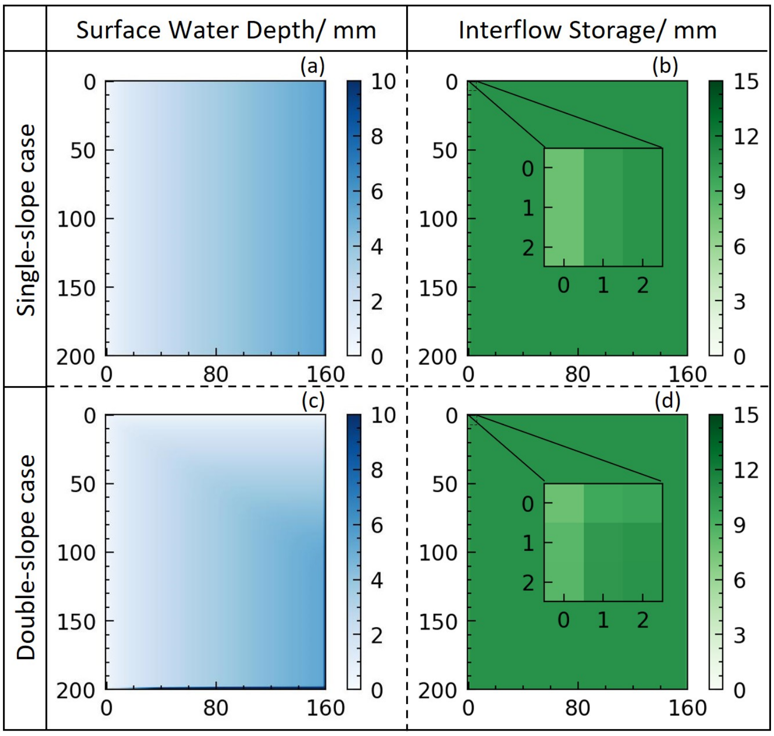

Figure 7. The spatial distribution of surface water depth (h_s) and interflow storage (O_i) on the left-side hillslope of both single-slope (a-b) and double-slope (c-d) synthetic V-catchment test cases at the 60 minute mark. The state variable distributions shown are simulated using two-dimensional (2D) slope concentration methods. The corresponding results of 1D methods are identical to those obtained from the single-slope case simulated with 2D methods, regardless of the test case used. For a clear comparison, the spatial distribution of O_i in the upper left corner has been zoomed."

Furthermore, we selected three points (A, B, and C, see Figure R7) from the zoomed area in the updated Figure 7. For both single-slope and double-slope cases, we generated time profile plots of the interflow storage using both the 1D and 2D linear reservoir method at these points (Figure R8). The comparative results clearly reveal the simulated differences between the 1D and 2D linear reservoir methods in the double-slope test case (Figures R8d to R8f).

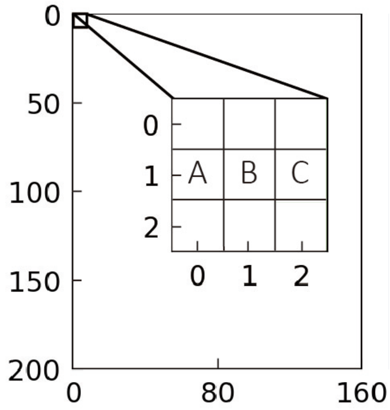

Figure R7. Diagram of the spatial location for points A, B, and C.

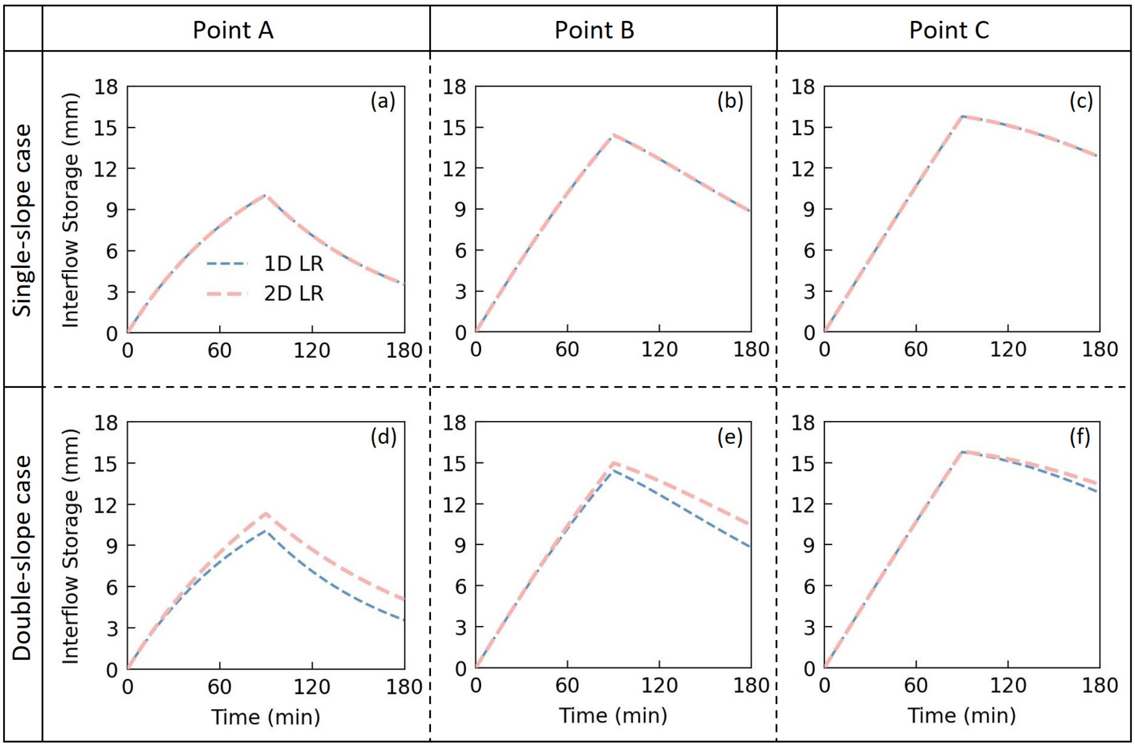

Figure R8. Timeseries of interflow storage (O_i) simulated by 1D and 2D linear reservoir (LR) methods for single-slope (a-c) and double-slope (d-f) synthetic V-catchment test cases (at points shown in Figure R7).

Below are some of my suggestions regarding the writing or presentation (although the paper is already well-written):

Specific issues:

1. Line 13-14: Please emphasize both the 2D diffusion wave and the 2D linear reservoir method in this sentence to enhance consistency with the paper.

Thanks for your suggestion. We will revise the sentence into:

“We also introduced two-dimensional (2D) diffusion wave equations for surface slope concentration and derived 2D linear reservoir equations for subsurface slope concentration, to replace their 1D counterparts.”

2. Line 58-59: Beven's alternative blueprint encompasses several key concepts, including Bayesian philosophy, model equivalence, and an emphasis on uniqueness. It would be beneficial to clarify which of these are closely related to the development of XAJ, although this may be considered a matter of fine-tuning.

Thanks for your suggestion. Beven’s alternative blueprint indeed encompasses several key perspectives, including those you mentioned, as well as an evaluation on the physical-based property of distributed hydrological model (Beven, 2002). Before the alternative blueprint, physical-based model generally emphasized on applying rigorous mathematical-physical equations deriving from established physical principles under appropriate assumptions, such as different simplified form of the Navier-Stokes equations. However, the alternative blueprint, while still adhering to fundamental physical principles and conservation laws, places greater emphasis on consistency with observational data. This shift effectively relaxes the strict dependence on mathematical-physical equations, enabling the integration of conceptual parameterization equations from the lumped hydrological model, thereby expanding the available equation space for distributed hydrological model. This laid the foundation for the development of the distributed version of the XAJ model, making it closely linked to the evolution of the XAJ model.

According to your suggestion, we will revise it to:

“Although initial efforts began earlier (Lu et al., 1996), 2002 was a notable milestone due to Beven’s alternative blueprint (Beven, 2002). This blueprint emphasized the importance of observational consistency as a core requirement of the physical-based property in DHMs, enabling the integration of parameterized equations from lumped hydrological models.”

3. Line 342: Please clarify the model's inputs and outputs, and whether they differ from the original XAJ model.

As suggested, an appendix will be included in the revised manuscript to clarify the inputs and outputs of the TDD-XAJ model, along with an analysis of whether they differ from the original XAJ model. The appendix is provided below:

“Appendix B: Inputs and outputs of the TDD-XAJ model

The inputs and outputs of the TDD-XAJ model are listed in Table B1. Its inputs consists of four categories: (1) model configuration dictionary, (2) spatiotemporal multidimensional arrays of precipitation and evaporation as forcing inputs, (3) raster datasets providing spatial attributes (including elevation, river network, aspect, etc.), and (4) lookup tables storing hydraulic parameters for surface and channel routing. The outputs comprise spatiotemporal distributions of state variables and model fluxes at pre-defined temporal resolutions (instantaneous or time-averaged values). Compared to the original XAJ model, which is a lumped hydrological model, its inputs are watershed-scale parameters and timeseries for precipitation and evaporation, with output mainly including simulated discharge hydrograph. The TDD-XAJ model share largely similar input and output information with existing distributed XAJ model, the aspect information introduced by the 2D linear reservoir method can be derived from elevation raster.

Table B1. Detailed information on input and output files of the TDD-XAJ model.

Class	Category	Name	Format	Description
Input	Dictionary	Model configuration	YAML	(1) Paths for the other files in this table, temp and output folders (2) Numerical calculation settings (e.g., CFL coefficient) (3) Model parameters
	Array	Forcing	NPY	Spatiotemporal distribution of precipitation and evaporation
	Raster	Elevation	Esri ASCII	Use projected coordinate system, with unit of m
	Raster	Aspect	Esri ASCII	Derived from elevation raster via GIS analysis
	Raster	Flow accumulation	Esri ASCII	Derived from elevation raster via GIS analysis
	Raster	Flow direction	Esri ASCII	Derived from elevation raster via GIS analysis
	Raster	River network	Esri ASCII	Mark the location of channel cells and corresponding code
	Raster	Land use	Esri ASCII	Aligned with elevation raster, and store land use code
	Lookup table	Surface hydraulics	CSV/XLSX	Land use code and corresponding surface roughness coefficient
	Lookup table	Channel hydraulics	CSV/XLSX	Channel code and corresponding attributes (e.g., longitudinal slope and channel roughness coefficient)
Output	Array	State variables	NPY	Spatiotemporal distributions of state variables at pre-defined temporal resolutions (instantaneous or time-averaged values)
	Array	Model fluxes	NPY	Spatiotemporal distributions of model fluxes at pre-defined temporal resolutions (instantaneous or time-averaged values)

”

4. Line 359: “The synthetic V-catchment, first proposed by (Overton and Brakensiek, 1970)”, This appears to be an incorrect citation format.

Thanks for pointing this out, we will revise it into:

“The synthetic V-catchment, first proposed by Overton and Brakensiek (1970)”

5. Figure 6: The lower-left corner should be "subsurface" rather than "surface."

Thank you for pointing this out, we will correct this.

6. Figure 8: Highlight the inflection.

Thanks for your suggestion. We have highlighted the inflection point in Figure 8 (a) using a red dashed box and plotted the first-order numerical derivative to emphasize this point. The updated Figure 8 is shown below:

"Line 519:

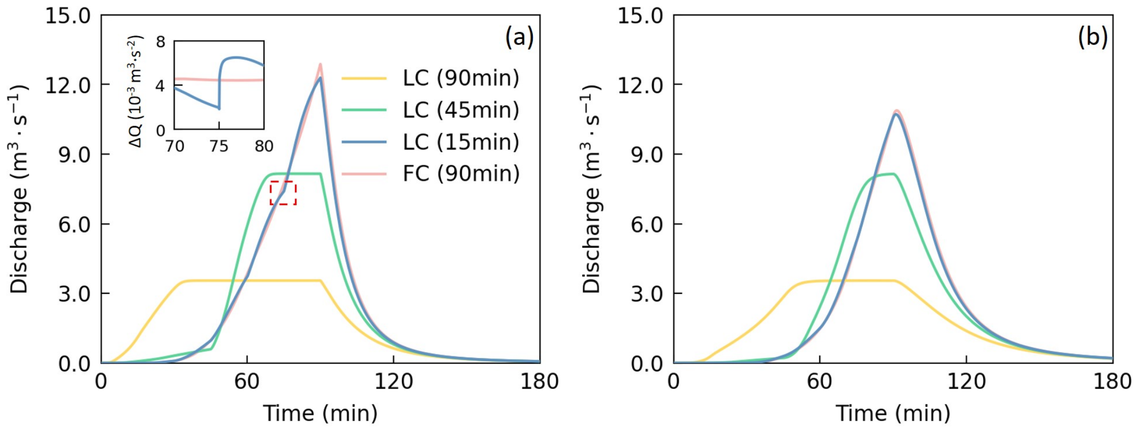

Figure 8. The simulated hillslope outflow (a) and channel outflow (b) of the loosely-coupled (LC) model and fully-coupled (FC) model in the double-slope case. The time in parentheses denotes the time interval of input forces (ΔT) used for the model. The FC model uses 90 min only, while the LC model uses 90 min, 45 min, and 15 min for convergence test. The parameters used are the first set of the parameters sampled with the SLH method from their ranges listed in Table 2. The inflection point at 75 minute mark in the hillslope outflow hydrograph of the LC model (using ΔT=15min) is highlighted with a red dashed box, along with timeseries of the first-order numerical derivative of hillslope outflow (ΔQ)."

References mentioned in the response

Beven, K., 2002. Towards an alternative blueprint for a physically based digitally simulated hydrologic response modelling system. Hydrological Processes, 16(2): 189-206.

Lam, S. K., Pitrou, A., and Seibert, S.: Numba: a llvm-based python jit compiler, in: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, Austin, Texas, 7, https://doi.org/10.1145/2833157.2833162, 2015.

Lu, M., Kioke, T., and Hayakawa, N.: Distributed XinAnJiang model using radar measured rainfall data,in: Proceedings of International Conference on Water Resources & Environmental Research: Towards the 21st Century, Kyoto, Japan, 1996.

Murphy, B., Yurchak, R., and Müller, S.: GeoStat-framework/PyKrige (1.7.2), Zenodo [code], https://doi.org/https://doi.org/10.5281/zenodo.11360184, 2024.

Overton, D. E. and Brakensiek, D. L.: A kinematic model of surface runoff response, in: Proceedings of the Wellington Symposium, Wellington, New Zealand, 1970.

Stacke, T. and Hagemann, S.: HydroPy (v1.0): a new global hydrology model written in python, Geoscientific Model Development, 14, 7795-7816, https://doi.org/10.5194/gmd-14-7795-2021, 2021.