rSHUD v2.0: Advancing SHUD and Unstructured Hydrological Modeling in the R Environment

Lele Shu[1, 2, 3], Paul Ullrich[2], Xianhong Meng[1,2], Christopher Duffy[4], Hao Chen[1,2,5], and Zhaoguo Li[1,2]

[1]Key Laboratory of Land Surface Process and Climate Change in Cold and Arid Regions, Northwest Institute of Eco-Environment and Resources, Chinese Academy of Sciences, Lanzhou, Gansu 730000, China

[2] University of Chinese Academy of Sciences, Beijing 101408, China

[3]Department of Land, Air, and Water Resources, University of California, Davis, Davis, California 95616, USA

[4]Department of Civil and Environmental Engineering, Pennsylvania State University, University Park, Pennsylvania 16802, USA

[5] College of Atmospheric Sciences, Lanzhou University, Lanzhou, Gansu 730000, China

Correspondence: Xianhong Meng (mxh@lzb.ac.cn) and Lele Shu (shulele@lzb.ac.cn)

Contents

Abstract. Hydrological modeling is a crucial component in hydrology research, particularly for projecting future scenarios. However, achieving reproducibility and automation in distributed hydrological modeling research for modeling, simulation, and analysis is challenging. This paper introduces rSHUD v2.0, an innovative, open-source toolkit developed in the R environment to enhance the deployment and analysis of the Simulator for Hydrologic Unstructured Domains (SHUD). The SHUD is an integrated surface-subsurface hydrological model that employs a finite volume method to simulate hydrological processes at various scales. The rSHUD toolkit includes pre- and post-processing tools, facilitating reproducibility and automation in hydrological modeling. The utility of rSHUD is demonstrated through case studies of the Shale Hills Critical Zone Observatory in the USA and the Waerma Watershed in China. The rSHUD toolkit's ability to quickly and automatically deploy models while ensuring reproducibility has facilitated the implementation of the Global Hydrological Data Cloud (https://ghdc.ac.cn), a platform for automatic data processing and model deployment. This work represents a significant advancement in hydrological modeling, with implications for future scenario projections and spatial analysis.

1 Introduction

Scientific modeling utilizes mathematical equations ~~of~~ representing natural laws to predict unknown variables in space and
time by incorporating known variables (Beven, 2012; Duffy, 2017). Hydrological modeling remains ~~a crucial approach in hydrology research~~ crucial for analyzing future scenarios and for for hypothesis testing. As hydrological models evolve from lumped to distributed models, the need for spatial data in modeling continually rises. Simultaneously, with the advancement of numerical methods as computation strategies in hydrological modeling, the models pose new challenges for hydrological unit partitioning (Paniconi and Putti, 2015; Peckham et al., 2017; Peel and McMahon, 2020).

Many hydrological models rely on Graphical User Interface (GUI) tools for data pre- and post-processing, such as arcSWAT (Arnold et al., 1998), PIHMgis (Kumar et al., 2010; Bhatt et al., 2014) and etc. Definitely, the GUI interface tools are user-friendly and easy to use, making them favorable for promoting models. However, there are two common challenges with GUI interface tools. First, poor repeatability due to difficulties in duplicating a user's data editing process leads to modeling discrepancies in the same simulation area. Second, GUI interface tools face difficulties in handling large amounts of modeling ~~. In watershed modeling~~ tasks. In modeling with GUI-tools, human participation is required instead of being controlled via modeling parameters, making it impossible to implement large ~~or automated modeling with GUI-based modeling tools~~number of hydrological modeling cases. For instance, the deep learning methods' modeling process, which is mainly completed automatically via parameter configuration, allows 671 basins in CAMELS data to be automatically modeled, optimized, and verified (Newman et al., 2017; Beven, 2020; Nearing et al., 2021; Li et al., 2022). However, similar automated modeling, simulation, and analysis work with a distributed hydrological model seems impossible mission; considering that lumped models such as SAC-SMA (Burnash et al., 1973), which are frequently compared with deep learning models in the research literature, require minimal data pre-processing.

Therefore, when dealing with a large number of watershed simulation tasks, both pre-processing and simulation post-processing ~~require~~ necessitate the utilization of intervention-free, reproducible, and automated tools. Hence, the development of automated and reproducible pre-processing and post-processing tools for distributed hydrological models and numerical method hydrological models is imperative. However, regardless of the modeling tools used, it encompasses epistemic uncertainty, affecting the final results and accuracy of hydrological simulations(Beven and Young, 2013; Beven, 2013, 2018, 2019). Various modeling tools can technically support the modeling process but cannot eliminate the inherent uncertainties in models and data. Users need to be vigilant about this.

This paper introduces the rSHUD tool, an open-source toolkit developed in the R environment and supports pre- and post-processing functionalities for the Simulator of Hydrological Unstructured Domains (SHUD) and similar surface-subsurface integrated hydrological models. Its tools can be combined into automated processing scripts suitable for batch modeling, simulation, and analysis tasks.

The article begins by presenting the R environment and the rSHUD package in section 2, followed by a brief introduction of SHUD model structure in section 3. Then the critical steps in deploying the SHUD model using rSHUD tools are described

in section 4. Lastly, in section 5, the paper showcases two basins as case studies to demonstrate rSHUD's modeling and results analysis process.

2 R and rSHUD

R, a freely available programming language, is widely utilized for data analysis and statistical computations. Renowned for its extensive core functions and libraries, R supports user-defined functions in multiple languages, including C/C++, Python, and Fortran. Its cross-platform compatibility encompasses Linux/Unix, Mac OS X, and Windows. RStudio (https://rstudio.com, accessed Sept. 2023) is a popular GUI for R. Beyond core libraries, R allows easy integration of additional libraries. Users can swiftly install these from the CRAN repository using *install.packages* or access developer libraries on GitHub via the *devtools:install_github* function. The *library* and *require* functions enable loading of these packages into the active environment.

rSHUD is an open-source project available on GitHub (https://github.com/shud-system/rSHUD, accessed June 2023) and regularly updated. Since it is not yet available in CRAN's repository, users can install rSHUD and the necessary libraries in their environment using the *devtools* library. These two commands install rSHUD and the required libraries in the user environment. The following will install the required libraries and rSHUD in the user environment.

```
install.packages("devtools")
devtools::install_github("SHUD-System/rSHUD")
```

The rSHUD version matches the SHUD model version. The current version of rSHUD is 2.0, designed to support SHUD v2.0 (Shu, 2023a). To ensure compatibility and streamline user experience, ~~The~~ the development team maintains concurrent versioning for both rSHUD and SHUD. While rSHUD is developed using the R programming language, SHUD is implemented in C/C++. The versioning process is managed manually to ensure consistency between the two.

Installation of rSHUD and the dependent library may take some time. The additional R libraries required for rSHUD are listed in Table A1 of appendix. The rSHUD package serves several important purposes in the field of geospatial data analysis and hydrological modeling. These include:

- Convert geospatial data to SHUD format. This package is equipped with a toolkit that handles raster and vector data. It then constructs an unstructured triangular mesh domain, which is an important step in preparing SHUD model data.

- Parameterization of the hydraulic properties of soil and land cover. This feature enables users to define and adjust the hydraulic properties of various soil types and land cover classes, improving the applicability of the model.

- The ability to read and write SHUD input files. This feature ensures smooth integration with the SHUD model and allows users to import and export data easily.

- The ability to load the output results generated by the SHUD model. This facilitates the interpretation and further analysis of the model's results.

- Facilitate hydrological time series and geospatial analysis. This feature allows users to perform detailed temporal and spatial analyzes of hydrological data, providing valuable insight into water dynamics.

- Exploring time series and spatial data. This function allows users to generate clear and informative visual representations of their data and helps with data interpretation and communication.

Each function in rSHUD has its own help page that provides information on its usage, arguments and return values. With more than 160 functions included in rSHUD, it is not feasible to provide explanations of all functions in this paper. However, users can access the help page for each function by using the command *help(FunctionName)*.

3 Model and Data

3.1 SHUD

SHUD is a distributed hydrological model that is based on physical principles, namely, the Saint Venant Equation for the surface runoff and Darcy-Richards Equation for the subsurface flow (Shu et al., 2020). It solves the partial differential equations (PDEs) of hydrology with the finite volume method (FVM), which allows for the direct coupling of equations representing groundwater, soil moisture, surface water, vegetation, and land cover interactions. The model's nomenclature, Simulator of Hydrological Unstructured Domains, stems from this intricate process of domain decomposition and the subsequent numerical solution of the associated PDEs within these unstructured hydrological domains. SHUD is an improvement and revision of the Penn State Integrated Hydrologic Model (PIHM) (Qu and Duffy, 2007; Kumar et al., 2010; Bhatt et al., 2014; Yu et al., 2014). Shu et al. (2020) provides a detailed explanation of the differences between the two models.

The SHUD source code, data for three exemplary watersheds, and a straightforward result analysis R script are available on GitHub (https://github.com/shud-system/shud, last accessed in August 2023) and as referenced in (Shu, 2023a). The three showcased examples include Qinghai Lake and the Heihe headwater in China, along with Cache Creek in the United States. Users can download the source code package, compile the model, and initiate the simulation. After the simulation, users can execute the provided R script within the RStudio GUI to retrieve simulation results, facilitating subsequent analysis and visualization.

SHUD offers flexibility in terms of time and spatial resolution. The spatial resolution of the model ranges from centimeters to kilometers, depending on the modeling needs and computational resources available. The internal time step, specified by the user as the maximum time step, can be adjusted, while the computing time step is limited to a few seconds. The model exports the status of the catchment at regular time intervals, ranging from minutes to days. Due to its flexibility, the model can be coupled with other systems such as agriculture, cryosphere, ecology and natural disasters. The SHUD model comprises two types of cells: hill-slope and river cells. In the planar view, the hill-slope cell has an unstructured triangular shape, while the river cell is portrayed as a rectangle. In the 3D view, the hill-slope cell is a triangular prism, and the river cell is a trapezoidal prism (Figure 1). Both the hill-slope and river cells are hydrological computing units (HCUs) in the SHUD numerical sover.
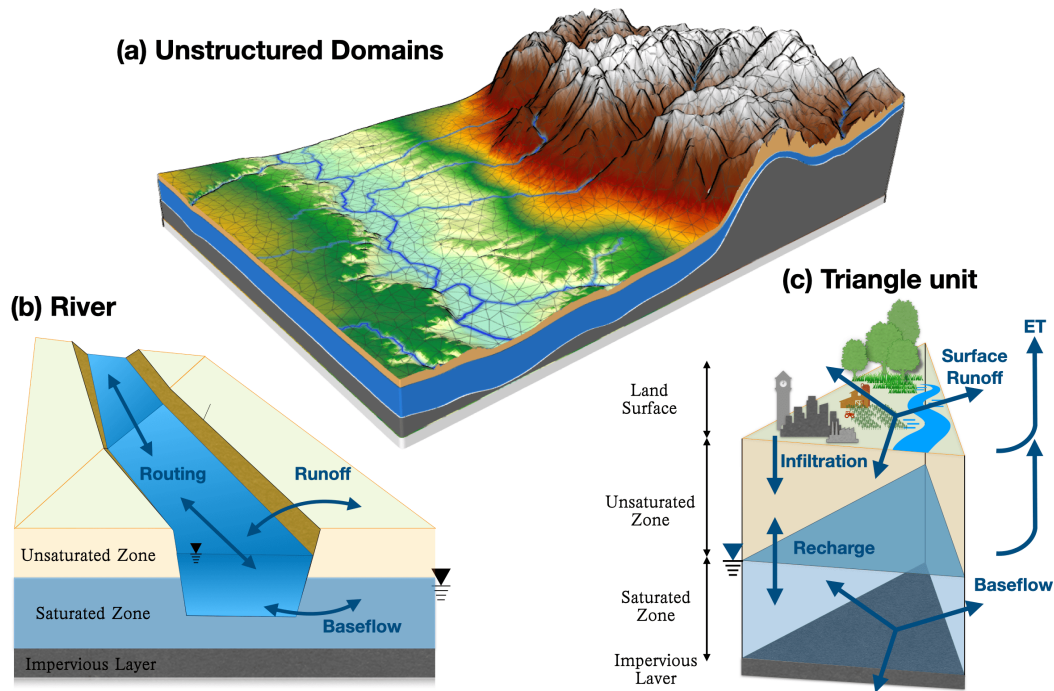
Figure 1. Spatial and hydrological structure of the SHUD model. Unstructured domains in the partial watershed (a), River cells and topological relationships with triangular cells (b), and 3-layer triangular prisms(c).

3.1.1 Hill-slope cell

The SHUD model utilizes unstructured grids to represent the computing domain of slopes. By default, the Denauley triangle
145   method is used for building the domain, although other triangular network generation methods are also acceptable. Each triangle comprises three nodes with unique three-dimensional coordinates (x, y, z in meters) that define their location. The centroids of the cells are calculated within the SHUD program. Topological relations are critical to the model and describe the one to three neighbors and nodes associated with each triangle.

For more detailed explanation and mathematical representation, readers can refer to Shu et al. (2020). We brief the four
150   crucial processes in the watershed hydrology :

  – Surface Water Partitioning: While Hortonian and Dunnian overland flows are common assumptions in conceptual models, Integrated Surface-Subsurface Hydrological Models (ISSHMs) like SHUD adopt a more physical description. Instead of these assumptions, they use the Darcy-Richards equation, such as the Green-Ampt method. In SHUD, surface runoff is calculated using Manning's formula for a hydrological computing units (HCU) .

155   – Evapotranspiration: ~~potential~~ Potential evapotranspiration (PET) is computed using the Penman-Monteith equation, while ~~Actual~~ actual evapotranspiration (AET) is derived by multiplying PET with a soil moisture stress coefficient, determined by soil moisture content and groundwater table depth.

   – Subsurface flow: Once water infiltrates the ground, it first moves vertically in the Unsaturated Zone. The flux from unsaturated zone to the saturated zone is termed as "recharge to groundwater." The calculation of horizontal groundwater
160   flux in ~~three dimensions~~ horizontal direction is based on the Dupuit Assumption.

   –

3.1.2   River cell

In geospace, a river network is a series of sequentially connected polylines that are defined as an ordered set of nodes in three-dimensional coordinates. A river reach is a polyline between two critical nodes. A critical node can be the intersection
165 of multiple rivers or a user-assigned point. Ordinarily, the first node marks the beginning of a reach, while the last node represents its end. The Strahler stream order (Strahler, 1952) in SHUD determines the order of branching of the river system. The triangular domain in SHUD intersect the river network, and the topological relationship between the polylines and triangles determines the exchange of surface and subsurface water between the river channel and hill-slope (Shu et al., 2020). The river outlets are typically located at the edge of the watershed.
170   The topological relationships between the river and hill-slope cells are an essential difference between SHUD and PIHM. In the PIHM model, the river network is adjacent to two triangular cells, which results in three issues:

1. The length of the river network has an important impact on the number of computing units in both river and hill-slope domain. Users need to balance river channel simplicity with the number of computation units; the simplicity implies modification on natural feature of river network.

175 2. In plain areas, the heavily meandering river network generates hundreds of small triangular cells, easily exceeding the necessary number of HCUs and slowing down model computation dramatically and unnecessarily.

3. The accumulation of water in sink cells violates the fundamental assumption of the shallow surface runoff equation, consequentially, cause the model to become unstable and result in poor performance. At the start point of first-order stream, the occurrence of local sink points are very often and greatly reduce the computational performance of the entire
180   watershed.

To tackle the above issues in PIHM, a modeler need to manually modify the shape of the river channels repeatedly, which reduces the efficiency and reproducibility of modeling. Accordingly, the river network of the SHUD model is superimposed upon triangular cells. This configuration facilitates the computation of groundwater exchange between hill-slopes and the river network, which is determined by the hydraulic gradients existing between the river channel and the groundwater level.
185 Meanwhile, surface fluxes are calculated utilizing the weir flow equation. (Shu et al., 2020).

3.1.3 Vertical layers

SHUD defines three vertical layers for each hill-slope cell (Figure 1(c)): the land surface, the unsaturated layer (vadose zone), and the saturated layer (groundwater). By default, the model assumes that the no-flow boundary at the bottom is an impervious layer, also known as the impervious bedrock layer.

Thus, the model's default settings define three elevations: the land surface (z_s), the groundwater table (z_g), and the bedrock (z_b). The water stored above z_s is called surface ponding water, while the water between z_g and z_b is known as groundwater. The elevations of z_s and z_b remain constant for each cell, whereas z_g fluctuates based on groundwater storage. The space between z_s and z_g is known as the vadose zone. If z_g rises enough to meet z_s, then the vadose zone disappears.

3.2 Rawdata

Hydrological models are a combination function that accepts certain input data and parameters and then produces their output. For modelers, the first question is: what data do we need for a hydrological model and what results do we get from the model?

The rSHUD package utilizes three primary types of data: spatial data, time-series data, and attribute data.

– **Spatial Data:** This encompasses a variety of geospatial elements such as elevation, soil classes, land cover classes, meteorological stations/coverage, the boundary of a research area, and the stream network. Spatial data can be in either vector or raster format, both of which effectively represent spatial heterogeneity.

– **Time-Series Data (TSD):** This includes meteorological data (precipitation, temperature, radiation, wind speed, and humidity) and phenological data (leaf area index and snow melt factor). Hydrologic models, depending on their conceptual and mathematical scheme, are sensitive to the time interval of the forcing (or meteorological) data. For instance, the intensity, duration, and frequency (IDF) of precipitation are critical to models for flood prediction, soil erosion, pollutant monitoring, and so forth. Therefore, for short-term hydrologic events, hourly to daily meteorological data is preferred in process-based models, while long-term trend modeling accepts daily to yearly data in most conceptual or water-balance models. Given that the SHUD model employs physical hydraulic equations and depicts hydrologic processes in fine spatial and temporal scales, it is recommended to use sub-daily meteorological data.

For calibration and validation purposes, the model also requires reference data, which could be observational data, comparable datasets from other models, or data from previous publications.

– **Attribute Data:** This includes the feature of the spatial data, used for generating hydraulic parameters of each soil class, geologic layers, and land cover classes. The required soil texture parameters include the percentage of silt and clay, organic matter content, and bulk density. These parameters are used to calculate the hydraulic parameters, including hydraulic conductivity, porosity, and the α and β values in the van Genuchten equations (Shu et al., 2020).

4   Model deployment

Deploying a hydrologic model involves several basic steps. Figure 2 illustrates the typical workflow of hydrologic modeling in a research region. First, data preparation is required to build a dataset subset for the research area. Second, data pre-processing is necessary to reformat the spatial data and attribute table. Third, the model must be built, generating input files for the hydrologic model. Fourth, the program must be executed to perform the modeling. Finally, post-processing is to read the output files and analyze the results.

The rSHUD package aims to support the three out of five steps in the model deployment: data pre-processing, building model and post-processing.
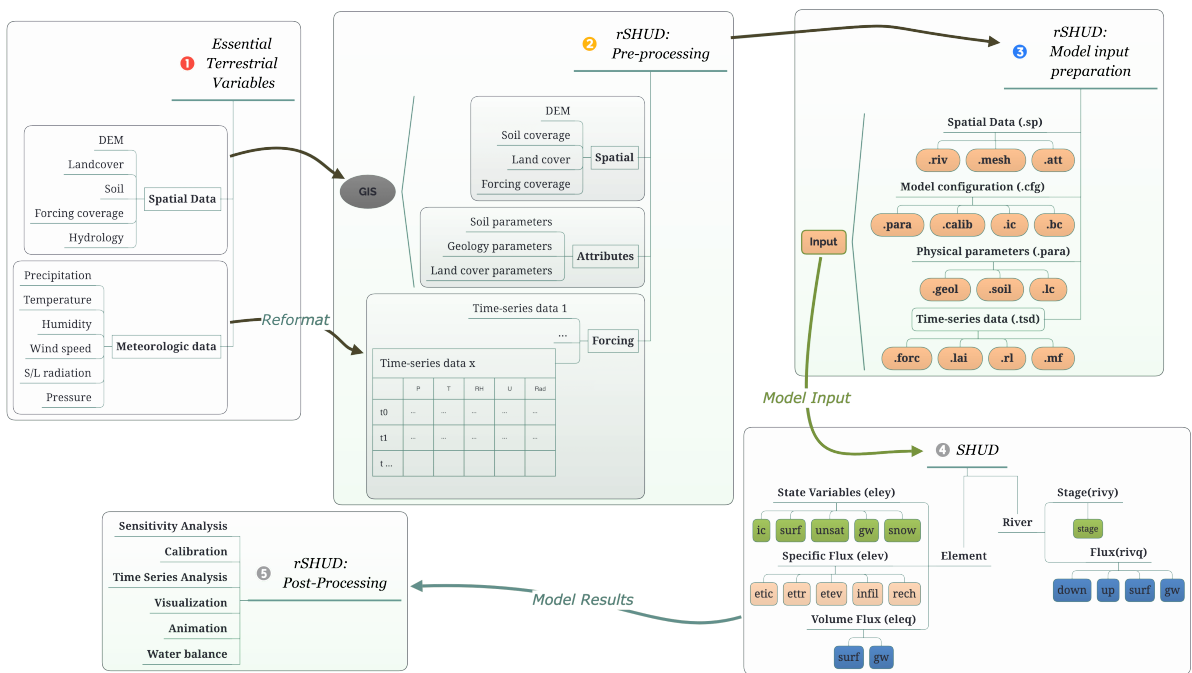
Figure 2. The general workflow of hydrological modeling and implementation of SHUD modeling system, which includes five steps: raw data accessing, pre-processing, building model

4.1   Data pre-processing

Multiple data processing stages were involved in this step. Holes were removed, modeling boundaries were projected, and buffer zones were generated in sequential procedures. Irrelevant data from the Digital Elevation Model (DEM) were excluded, retaining only pertinent information within the study area. The DEM data underwent reprojection and simplification into a Projected Coordinate System to facilitate analysis. The river flow direction consistency for the river network data was verified and corrected, while duplicate points and segments were eliminated, and the data format was standardized.

The data pre-processing stage included reformatting the spatial data into a consistent format and generating hydraulic parameters based on the land cover, soil, and geology classification. Prior to data pre-processing, sufficient spatial data were ready, including DEM, soil, land cover, watershed boundary, and river network. As the SHUD model requires a specific format for the time-series data, the forcing and phenology data format and units must be standardized. The data type of time series in R is *xts*. The *xts* data class supports a matrix format, where rows represent time and columns represent multiple variables. For example, when saving meteorological forcing data, the columns of the data represent precipitation, temperature, wind speed, radiation, and air pressure. Regardless of the original format of the user's data, it can be saved as the data format required by the SHUD model through the *write.tsd* function. The output file of the *write.tsd* function is self-explanatory, and users can use other software or programs to save it in other time series data formats.

Consistency of spatial coordination in data processing is indispensable to ensure accurate and reliable results. It is necessary for spatial data to have uniform projection information, which basically means a defined coordinate system. Analysis tools usually accept data either from a Geographic Coordinate System (GCS) or a Projected Coordinate System (PCS), a method to represent GCS on a flat surface. However, for hydrological modeling, data in different projections should be re-projected into a specific PCS before spatial processing. This is important since crucial spatial information from maps, including distance (in meters), direction, and area (in square meters), varies across different PCS. It is necessary to ensure the accurate overlay of data from multiple sources to prevent potential spatial inconsistencies.

It is recommended to use the Albers Equal Area projection, which has two reference latitudes and one central longitude. The reference latitudes are set at 1/4 and 3/4 of the watershed latitude range, respectively. Additionally, the central longitude corresponds to the longitude of the watershed centroid. An example is provided in section 5.

Typically, the extent of the raw spatial data exceeds the watershed boundary; therefore, it is necessary to subset the data. Moreover, the final boundary of the modeling domain may differ from the original watershed boundary because spatial processing simplifies it. Thus, the subset should be slightly larger than the research area. A enclosure mask layer is generated from the watershed boundary with a buffer distance.

The original attributes in the soil data include soil texture components such as silt, sand, and clay percentages, as well as bulk density and organic matter content. To derive hydraulic parameters such as hydraulic conductivity, porosity, and van Genuchten parameters, a pedotransfer function is used based on the soil texture data (Wösten et al., 2001; Shu et al., 2020). Pedotransfer functions, being empirical equations derived from specific regional laboratory data, inherently possess limitations in their universal applicability. Users have the flexibility to select and implement alternative PTFs. The primary value of PTFs in rSHUD is to offer an initial estimation of essential hydraulic parameters, while uncertainties in these parameter values should be considered. The pedotransfer function used in rSHUD is listed in Appendix C.
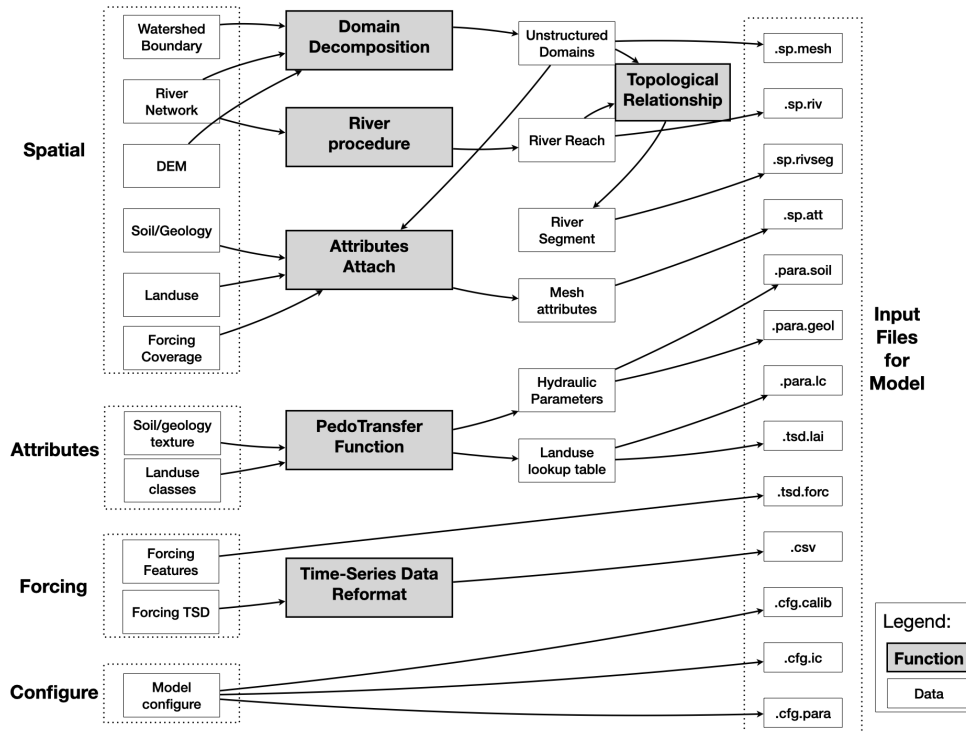
Figure 3. The procedures of model building and the outputs files of rSHUD. The left dash boxes are input rawdata and the right dash box are model input files.

4.2 Model build

4.2.1 Domain Decomposition

Domain decomposition is a mathematical strategy that employs geometric principles to partition a larger domain into smaller, more manageable subdomains. Within the context of the SHUD model, this approach is implemented to decompose the watershed into a Triangular Irregular Network (TIN). As a result, the watershed is composed of a collection of unstructured triangles.

The SHUD model can use regular triangles as the computational domains, but unstructured domains are recommended. Unstructured domains offer more flexibility to represent the irregular watershed boundary and terrain features. Moreover, unstructured domains allow for different resolutions within a watershed. For example, modelers can apply coarse TINs in the whole watershed but finer TINs in the river corridor when the research topic concerns surface-subsurface water exchange along the corridor. The multi-resolution configuration ensures sufficient resolution in the area of interest while maintaining the mass balance of the watershed, without increasing the number of HCUs excessively as in regular grid domains.

The triangulation method *shud.triangle* in rSHUD package is from *RTriangle*, an R port of Jonathan Shewchuk's Triangle library (Shewchuk, 1996, 2002). The *RTriangle* generates Delaunay triangles without no small or large angles and is thus suitable for finite element/volume analysis. The *shud.triangle* function requires watershed boundary as the mandatory input argument while DEM, river network, islands/holes and suggesting points are optional conditions to constrain the triangulation.

The additional spatial data inserted during the triangulation process allows users to adjust the resolution of HCU in different areas according to their research needs and study area characteristics, enhancing the flexibility of domain decomposition. There are three extra arguments in *RTriangle::triangulate()* that are useful to build the ideal domains: minimum angle of a triangle (argument *q*), maximum of triangle's area (argument *a*)and the ideal number of triangles(argument *S*).

Prior to performing triangulation, the boundary must be simplified using a tolerance. This process aids in smoothing the watershed boundary while ensuring an appropriate number of HCUs. For instance, when applying watershed delineation methods to retrieve the boundary of a watershed using a 30-meter DEM, the resulting boundary often exhibits jagged 30-meter scale edges. Therefore, in triangulation, the maximum edge of a triangle on the boundary is limited to 30 meters, while the inner portion of the mesh domain is composed of larger triangles. Consequently, ~~The~~ the substantial difference in triangle area between the edges and inner parts significantly slows down model performance by unnecessarily increasing computations on the boundary. This negative impact on model efficiency must be addressed by simplifying the boundaries. Simplification is also necessary for additional triangulation constraints such as a river, lake, and urban areas.

After triangulation, the *shud.mesh* function generates a mesh file that defines triangles by the index of three vertices and three neighbors and the definition of all vertices in the domains (x, y, z coordinates). The definition of TINs can be converted into a text mesh file by *shud.mesh*, which inversely can then be interpreted into a Shapefile of polygons by a GIS function called *sp.mesh2Shape*. These functions can cross-validate the consistency of the mesh domain.

4.2.2 River

The processing for the river is to 1) simplify the river reaches, 2) build the flow path, 3) determine the river order, 4) extract the slope characteristics and 5) describe the geometry of river reach. These functions are integrated into the *shud.river* function, the functions also can be called independently.

1. The simplification of the river reaches includes two meanings: the one is to simplify the meandering rivers straight, which is optional in SHUD; the other is to cut the very-long river reaches into smaller pieces.

2. In GIS and geomorphology, building a river path involves building the connections between upstream and downstream river lines. These connections form the river network and represent how water flows along the river lines. The direction of the river lines determines the inflow and outflow between two rivers. The function is realized by combining *sp.RiverDown* and *sp.RiverPath*.

3. River order is a categorization method used to understand the structure of a river network based on the levels of branching within it. The Strahler method (Strahler, 1952) is a wide approach for calculating river order. It assigns a value of 1 to all headwater streams and then increments the value by one for each confluence of two streams of equal order. In cases

where two streams of different order join, the ~~order of the downstream stream~~ higher of the orders of the two joining streams is used. This process is repeated downstream until all streams have an order assigned to them. The calculation of river order based on the Strahler method is performed by using the function *sp.RiverOrder*.

The order of river reaches determines the generalized geometry (definition of the trapezoidal shape) and hydraulic parameters (Manning's roughness, Chezy coefficient, and so on) of each river reach, since the strong co-relationship exist among slope, cross-section geometry, river length, contribution area, river order, discharge and so on (Flint, 1974; Kratzer et al., 2006; Downing et al., 2012; Perron and Royden, 2013; Strick et al., 2018; McManamay and DeRolph, 2019). In the default configuration, all river reaches in the same order share the same geometry and hydraulic parameters. When a detailed description of individual river reaches is available, the model can also accept a detailed description for each river reach in the model domain instead of categorization by river order.

4. The slope is a critical parameter used to calculate the fluxes in river routing. It is determined by calculating the gradient between the elevations and distances of the starting and ending points of a river reach.

5. The default geometry of a river cross-section is represented by a trapezoidal shape, defined by the width of the riverbed (w), the slope of the bank (s_b), and the depth of the river channel (D) (Figure 4). The trapezoidal shape is very flexible and can be simplified to a shape of a rectangle ($s_b = 0$) or a triangle ($w = 0$), depending on the specific case. The hydraulic parameters of the river channel encompass five components: sinuosity coefficient of the river reach, Manning's roughness, weir flow coefficient, conductivity and thickness of sediment. These parameters are pivotal in dictating open channel flow and the exchange dynamics between the hill-slope and the river reach. Given the inherent uncertainties and the lack of precise values during the initial model deployment, the *RiverType* function provides default values as an initial guess. However, it is worth noting that users retain the flexibility to modify these values based on their measurements or other reliable sources.
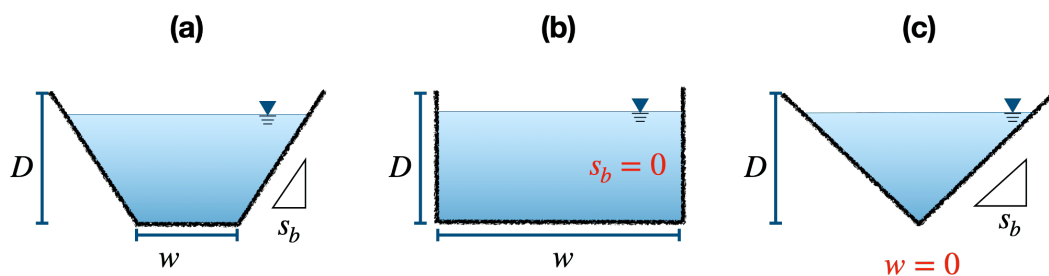
Figure 4. The cross-section of river in the SHUD model, which is described with the width of the riverbed (w), the slope of the bank (s_b), and the depth of the river channel (D). The trapezoidal shape (a) can be simplified to a shape of a rectangle (b) or a triangle (c).

The SHUD model determines the surface/subsurface water fluxes connection between the river and hill-slope cells based on their intersectional relationship. However, the topological relationship between triangles and rivers does not match perfectly.

Specifically, a river reach intersects with multiple triangles and exchanges water with these triangular cells. Therefore, to account for the exchange of water, each river reach is divided into several segments, each exchanging water with one triangular cell. The properties of each river segment include the index of the belonging reach, the index of the intersectional triangular cell, and the length of the segment within the cell. The function *sp.RiverSeg* is used to carry out this task.

4.2.3 Cell Attributes

The attribute file contains information about the features of each triangular cell, which is used for hydrological parameters. An example can be found in Figure 5. The index of soil is used in the *.att* file to specify a row of parameters in the *.para.soil* file. Similar to the soil index, the index for geology, land cover and forcing in the attribute file points to a row in the *.para.geol*, *.para.lc*, and *.tsd.forc* files respectively.
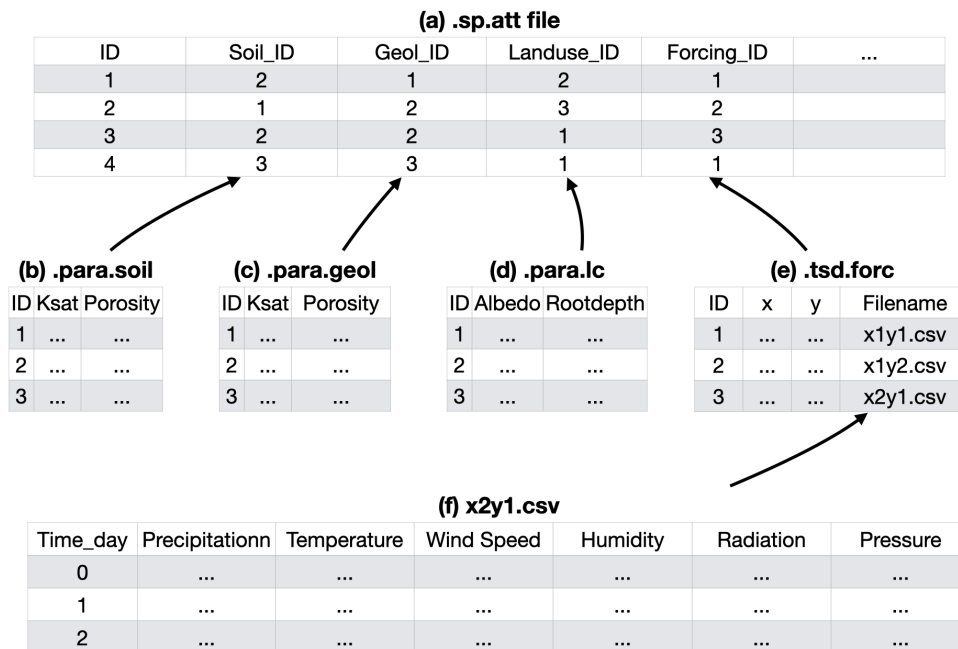
Figure 5. The structure of the model input files and their logical connections. Files are (a) attributes file (.sp.att), (b) hydraulic parameter of soil (.para.soil), (c) geology (.para.geol), (d) land cover (.para.lc), (e) time-series files of all forcing data (.tsd.forc), and (f) the time-series data for specific sites (.csv)

The *shud.att* function extracts the spatial data index as the properties of each cell. It extracts the features of multiple layers through the centroids of the triangular cells, thus capturing only the value of the centroid instead of the mean or statistical value of the entire cell. For example, a cell ~~may contain~~ contains only one soil type ~~with no~~ without heterogeneity within the cell.

The diversity of soil properties among cells expresses the hydrological heterogeneity in a watershed. Hence, to represent high heterogeneity, a finer resolution of the domain decomposition is required.

4.2.4 Hydraulic parameters

Three main files provide the necessary hydraulic parameters for hydrological simulation: soil, geology, and land cover. The pedotransfer function (Wösten et al., 2001) utilizes soil texture to derive the hydraulic parameters, including vertical and horizontal hydraulic conductivity and porosity, for the deep groundwater layer. The Appendix C lists the equations of the pedotransfer functions developed by Wösten et al. (2001).

1. The term "soil" specifically refers to the top layer of soil that influences the infiltration from the surface to the soil and the deep recharge to the saturated layer. The soil layer's essential physical parameters include vertical saturated conductivity, porosity, residual water content, and values α and β for the van Genuchten equation (Shu et al., 2020).

2. The SHUD model's geology layer refers to the aquifer profile's deeper layer. This layer's parameters describe the properties of saturated groundwater flow, and the crucial parameters include vertical and horizontal hydraulic conductivity and porosity.

3. The hydraulic parameters of land cover, such as vegetation root depth, soil degradation ratio, imperviousness factor, and Manning roughness, are stored in a lookup table within the data repository and can be easily transferred to a user's dataset for their modeling efforts. However, it is necessary to note that the lookup table is specifically designed for a particular land cover classification and is not transferable between different classification systems. For instance, the table for National Land Cover Datasets (Wickham et al., 2020) is not applicable to USGS 0.5-km MODIS-based Global Land Cover Dataset (Broxton et al., 2014) or China Land Cover Dataset (Yang and Huang, 2021). Appendix B provides more insight into the default values for multiple land cover classification systems. The University of Maryland (UMD) Global Land Cover Classification (Hansen et al., 2000) determines the parameters for typical land covers based on which the values for other classification systems can be adjusted and transferred. The parameters other than UMD are transferred from UMD data(Hansen et al., 2000; Bhatt, 2012). However, the values in the tables are somewhat arbitrary and act as preliminary values that need to be updated when users have more reliable data. Users can modify the values in the provided look-up tables as needed. If they are employing a different classification system not covered by our default tables, they would need to develop a new look-up table tailored to that system.

The full list of parameters in the SHUD model is list in appendix

4.2.5 Time-Series Data

The *.tsd.forc* file (Figure 5) saves a table about the forcing sites, including the x, y, and z of the sites and the time-series filename of it (Figure 5(f)). The SHUD program reads the time-series file during the model simulation.

The phenology in SHUD is LAI's TSD, indicating growth, prosperity, and withering. The TSD is also a experiential value for each land cover class from UMD vegetation parameter values(https://ldas.gsfc.nasa.gov/nldas/vegetation-parameters, accessed June 2023). Similar to the lookup table of land cover parameters, the LAI TSD can be replaced by the user's local observation data.

As the melting factor in SHUD is empirical monthly values for the Degree-Day model calculating the snow melting flux (Hock, 2003; Zhang et al., 2012), the function of *MeltingFactor* adapt the following equation from Bhatt (2012):

$$M_{\mathrm{f}} = \left\{ \left(\frac{M_{\max} + M_{\min}}{2} \right) + \sin\left(\frac{2\pi N}{366} \right) \times \left(\frac{M_{\max} - M_{\min}}{2} \right) \right\} \times 0.4 \tag{1}$$

M_{f} is the melting factor used in the Degree-Day model [$mm \ day^{-1} K^{-1}$]. The maximum and minimum values of the melting factor during a year are represented by $M_{\max}$ and $M_{\min}$, respectively, and occur on June 21st and December 21st. N reflects the Julian day difference from September 21st. As a result, the melting factor follows a sine curve, with its maximum on June 21st and minimum on December 21st. The denominator 366, which represents a leap year, is replaced with 365 for a common year.

Besides, there are boundary conditions (*.tsd.bc*) and observation data (*.tsd.obs*) in TSD format, which is optional for the model simulation. The boundary condition may be the irrigation, pumping, leaking, or known water management practices in time series. The optional observation data (*.tsd.obs*) is generally used for the model calibration only.

4.2.6 Model configure

In addition to the three primary categories of data – spatial, time-series, and attribute data – that hydrologic models commonly require, specific models may also necessitate the inclusion of configuration files. These files define various aspects of the model's operation, establishing the model's running range, determining the level of computational precision, setting the computing/exporting time-step, and specifying the file format, among other parameters. Thus, these configuration files provide a means to customize the model's functionality and output to meet specific research tasks and objectives.

The SHUD configuration includes four files: configuration of simulation (*.cfg.para*), calibration file (*.cfg.calib*), the initial condition of simulation (*.cfg.ic*) and boundary condition index (*.cfg.bc*). To generate boundary conditions with *shud.ic()* function, it is necessary to know the index of triangular meshes and river reaches. The indexes in boundary condition point to specific indexes in the TSD BC files. The initial conditions for river reaches include the initial water level in the channel. Details and file structures of these files are described in the SHUD manual (https://shud.xyz/book_en , accessed June 2023) (Shu, 2019).

5 Examples

To demonstrate the workflow of using the rSHUD package, we chose two watersheds as case studies and gradually implemented the processes of data pre- and post-processing of results. We selected the Shale Hills Critical Zone Observatory (SHCZO) in

the USA and the Waerma Watershed in China. The rSHUD package already includes all the raw data needed for building the hydrological model, so there is no need to download extra files.

The R scripts for these exemplary watersheds can be found in the appendices D1, D2, and D3. As the SHCZO is a small and simple catchment, we created a script for the deployment of the SHUD model briefly. The other two scripts (Appendix D2, and D3) are relatively sophisticated for pre- and post-processing the SHUD modeling in the Waerma Watershed. All these scripts are embedded in the rSHUD package already. The files *demo_sh.R* and *demo_waerma.R* in the rSHUD source code is used to deploy SHUD in the SHCZO and Waerma, respectively, and *demo_waerma_ana.R* is used for the post-processing of the Waerma Watershed. The R scripts are self-explanatory, and users can read the annotations and understand the functionality of the codes; Therefore, we have omitted the details in this paper.

5.1 Shale Hills Critical Zone Observatory in the USA

The Shale Hills Critical Zone Observatory (SHCZO) is a small ($\approx 84,000\ m^2$), forested catchment located in central Pennsylvania, USA. Its topography boasts relatively steep slopes (>0.18) and narrow ridges leading to its Shaver Creek tributary. The elevation of the catchment varies from 250 to 320 meters, and it experiences a humid continental climate that averages at 9.5 $°C$. SHCZO receives an annual mean relative humidity of 65.2% and precipitation of around 1092 mm. Evapotranspiration is estimated to be 662 mm with an annual runoff of 442 mm, translating to a runoff ratio of about 40-50% ((Jin et al., 2011; Shi et al., 2013; Brantley et al., 2018)).

All data for the SHCZO modeling is downloaded from the Critical Zone Observatory website (https://czo-archive.criticalzone. org/shale-hills/data/, accessed June 2023). The DEM is 1-meter LIDAR data, and soil is from the local survey (Lin, 2006; Yu et al., 2014). The watershed boundary and river network is calculated from the watershed delineation algorithm in PIHMgis (Bhatt et al., 2014). The local meteorological station provides the forcing variables.

Due to the availability of high-resolution data in this SHCZO and the watershed's small size, a high-resolution SHUD model will be constructed. Since the SHUD model implements a triangular mesh, the triangle's dimensions will vary. We anticipate a maximum triangle area of approximately 200 m^2. In the result, the function *shud.triangle* with the constraints produced 698 triangles in SHCZO (Figure 6). The area of cells within the mesh followed a normal distribution, with values ranging between 40 and 200 m^2. This distribution highlighted the presence of both small and larger triangles within the model.

The remaining R script of *demo_sh.R* requires no elaboration as it is easy to read. The code saves all data in a user-specified path. Once data preparation is complete, the SHUD model can be compiled and run. We named the SHCZO project as *sh*; to initiate the simulation on Linux-like platforms, the user should enter *./shud sh*. In this example, the forcing data covers two years (2008-2009), and the simulation took about 5 minutes to complete on an Intel 6-Core I5, 64G RAM computer. While the SHCZO model output closely mirrors the Waerma watershed visualization, appendix E displays the SHCZO simulation results. This paper's focus is on showcasing rSHUD's capabilities, so detailed parameter optimization for closer alignment with observations was omitted. Consequently, the simulation results serve primarily as a testament to rSHUD's functionality rather than for in-depth analysis.
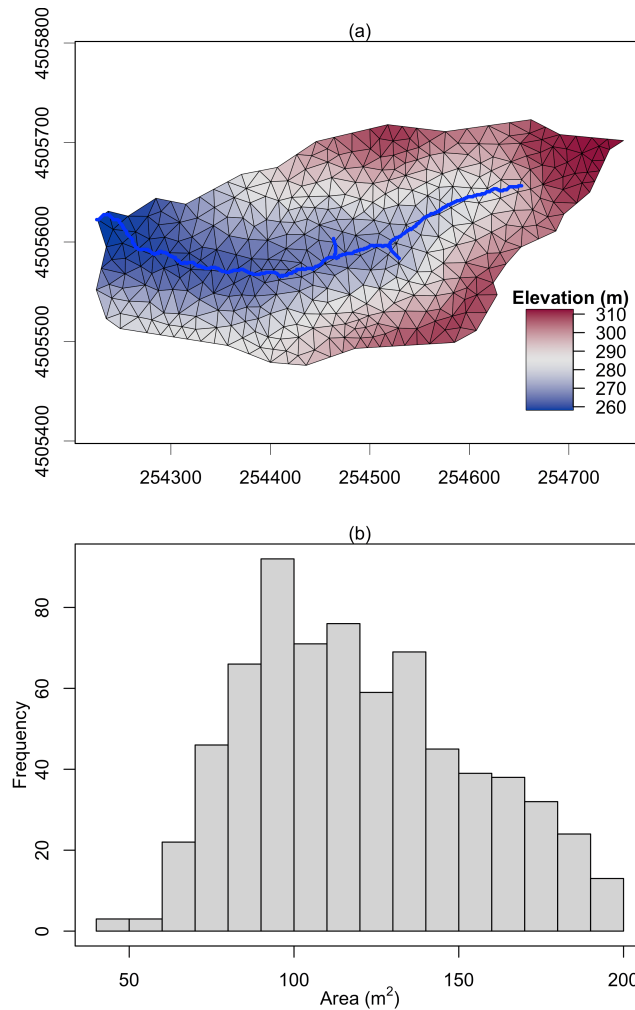
Figure 6. The triangular mesh (a) generated by the function *shud.triangle* for Shale Hills Critical Zone Observatory (SHCZO), and the histogram of triangles' area (b). The color in plot (a) is the centroid elevation of triangles.

5.2 Waerma in China

The Waerma watershed is a headwater of the Yellow River, with an area of 9.8 km^2, located near Waerma Village, about 20 kilometers northwest of Maqu County in Gansu province of China. It has an elevation of 3800-4500 m with significant terrain fluctuations and an average slope of about 0.42 (rise versus distance). The annual average temperature is about 1.2 $°C$, and the annual average rainfall is about 630 mm. The main vegetation types are grasslands, meadows, and shrubs. The Key Laboratory of Land Surface Process and Climate Change in Cold and Arid Regions, Chinese Academy of Science, built a comprehensive and detailed observational system for meteorology, land process, hydrology and cryosphere in the Waerma watershed (Meng et al., 2023).

Data	Description	Source
DEM	$30m$	ASTER Global DEM (NASA et al., 2018)
Watershed Boundary	-	Generated from DEM delineation
River Network	-	Generated from DEM delineation
land cover	$0.5\ km$	USGS MODIS land cover data (Broxton et al., 2014)
Soil	$1\ km$	Harmonized World Soil Database v1.2 (Nachtergaele et al., 2008)
Forcing	0.1 deg, 3-hr interval	China Meteorological Forcing Dataset (CMFD) (He et al., 2020)

Table 1. The description and source of rawdata for Waerma Watershed.

440   The expected modeling configuration is the following: using the CMFD data 2000-2001 to drive the SHUD model with larger than 150 m spatial resolution. The maximum triangle area was set to $150m \times 150m = 22500m^2$, therefore, there are a line *a.max = 150*150* in the R script (*waerma.sh*). The rawdata is described in Table 1. These data also can be retrieved via the Global Hydrologic Data Cloud (https://ghdc.ac.cn, accessed Sept. 2023) (Shu, 2023c).

5.2.1   Deployment

445 The script to deploy SHUD in Waerma Watershed is saved in *demo_waerma.R* and the watershed data is also available in the source code package. Following are the steps to deploy the SHUD model in Waerma Watershed by rSHUD.

1. Load necessary R libraries.

2. Setup and create folders for exporting data and figures.

3. Setup the environment for rSHUD.

450 4. Read/load this project's raw spatial and attribute data. All the spatial data must be reprojected to an identical PCS in this step.

5. Configure the modeling parameters, including the maximum triangles' area, the minimum angle of triangles, tolerance to simplify watershed boundary and river network, thickness of the aquifer, and number of days of simulation periods.

   The expected minimum resolution of modeling in Waerma is 150 m; therefore, in triangular mesh, the maximum cell
455   area is about 225000 m^2. After the domain decomposition with *shud.triangle*, 727 triangles are generated (Figure 8) and mean area of them is 13544 m^2 (equivalent to 116 m horizontal resolution).

6. Time-Series data processing, including the forcing data, LAI and melting factor. Also, the Thiessen Polygons of forcing sites are generated, which tell the matching TSD for each cell. The Thiessen polygons are not utilized for spatial interpolation of meteorological data. Instead, they are employed to delineate the coverage area for each meteorological station.
460   For instance, if we assume there are N triangles falling within the coverage of the 1st Thiessen polygon, then the *.sp.att*

file (5a) will assign a *Forcing_ID* of 1 to these N triangles. This indicates that the meteorological forcing data for these N triangles are provided by TSD of the 1st Thiessen polygon.

7. Attaching the attributes of soil, geology, and land cover to the triangular mesh.

8. Building the topological relationship between rivers and triangles.

465 9. Generating the model configuration files.

10. Writing the model input files out.
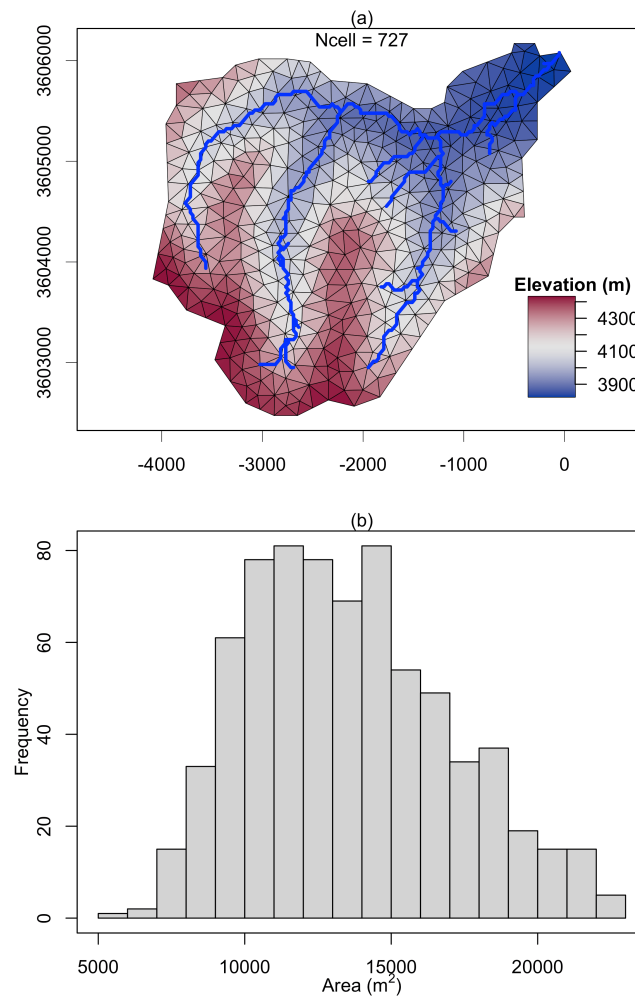
Figure 7. The triangular mesh (a) generated by the function *shud.triangle* for Waerma Watershed, and the histogram of triangles' area (b). The color in plot (a) is the centroid elevation of triangles.

5.2.2 Result Visualization

Once the simulation is complete, we can analyze the results (Shu, 2023b). The script of visualization *demo_waerma_ana.R* still needs to repeat the first three steps of loading R libraries, setting up folders, and the environment.

470   The *shud.env* function configures the global variable environment, setting up several default variables for data processing to boost post-processing. The input arguments of *shud.env* include the project name, the path of model input, and model output.

     After that, we start to load and plot the simulation results. A series of reading functions are available to read the model input files as well as the output files. The readout function reads the simulation results and returns multi-column time-series data, where the index for each column represents the index of HCUs. For example, the TSD for the jth river reach can be found in

475 column j of stream flow data (*.rivqdown*) , whereas the TSD for the jth triangular cell can be found in column j of the potential evapotranspiration file(*.elevetp*).

     Figure 8 demonstrates the visualization of the hydrograph (precipitation versus discharge), water balance (Storage change = Precipitation - Evapotranspiration - Discharge), the spatial distribution of groundwater table and annual mean evapotranspiration. Without calibration with observational data, this figure is outputs of preliminary simulation and the resulting values

480 may not be effective. Since the script to read and plot results are self-explained, users can read and modify the code based on their own needs. The script of model deployment and result visualization demonstrates the capability of rSHUD for pre- and post-processing of SHUD modeling.

6   Conclusion

The rSHUD is a toolbox developed in the R environment that supports the pre- and post-processing for the SHUD model.

485   The rSHUD package provides a set of tools to facilitate the conversion, parameterization, integration, analysis, and visualization of hydrologic data for the SHUD model. The package includes a toolkit for raster and vector data to construct unstructured triangular mesh domains. It also enables defining and adjusting hydraulic properties for soil types and land covers. The package ensures seamless integration with the SHUD model, with the ability to read and write input files and load output results. The package also enables detailed temporal and spatial analyses of hydrologic data and data visualization for easier interpretation.

490   Uncertainty is crucial in hydrological modeling and must be considered even when using the rSHUD package for model deployment. Users should acknowledge uncertainties in data inputs and model parameters, which may arise from measurement errors, natural variability, or limitations in the package's model structure and parameterization. The equations and data embedded within rSHUD package also introduce uncertainties. Users of the rSHUD package should therefore conduct a thorough uncertainty analysis as a preprocessing step to ensure the reliability and robustness of their modeling outcomes.

495   The tools in rSHUD not only boost the model deployment and analysis for the SHUD model but also can be used for other spatial analysis and hydrological data processing. The package has more than 160 functions developed in R and keep growing. Users can type command *ls("package:rSHUD")* to see a list of all functions available within the rSHUD package and *help(FunctionName)* to access the function description. An automatic data processing and model deployment platform, Global Hydrological Data Cloud (https://ghdc.ac.cn, accessed Sept. 2023), was implemented with the support of the rSHUD package.
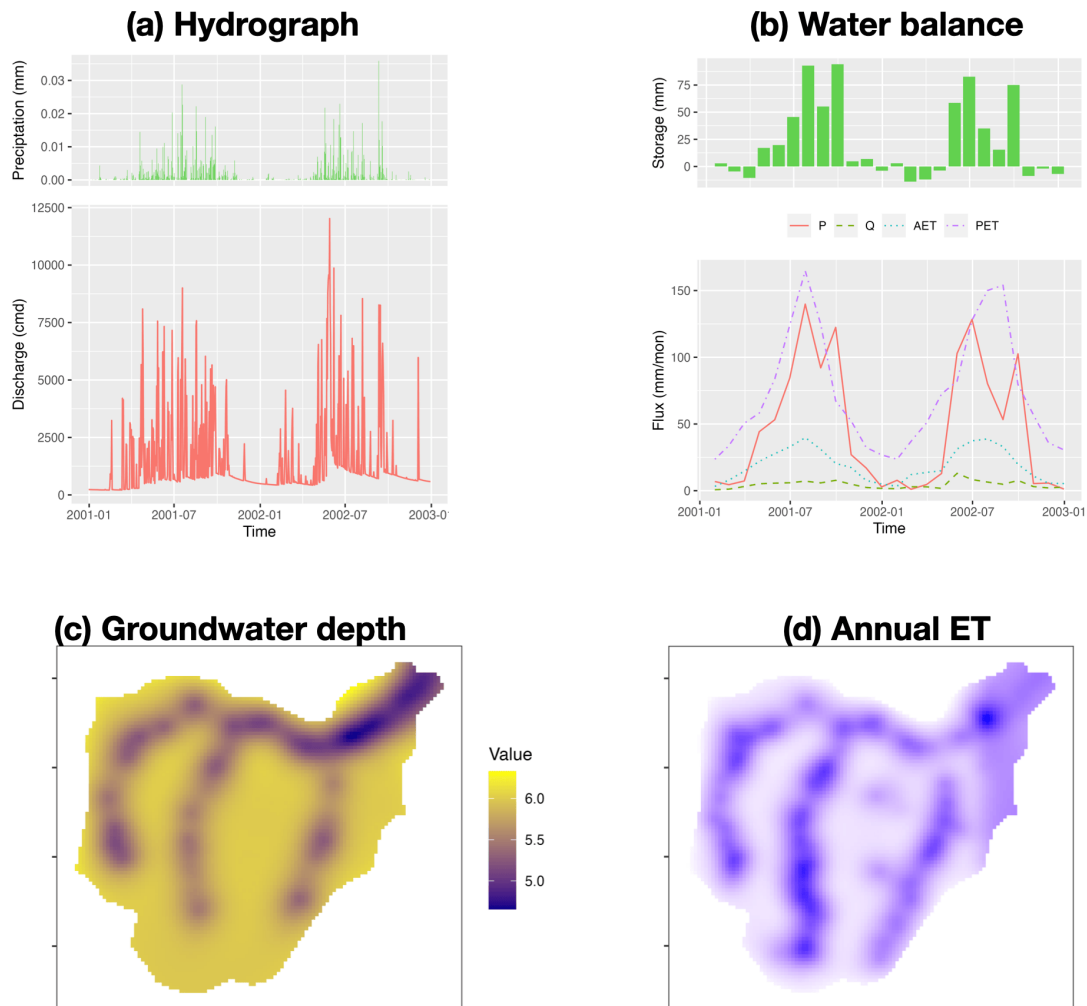
Figure 8. The result analysis of modeling Waerma Watershed; (a) hydrograph, (b) watershed scale water balance, (c) ground water depth [m], and (d) annual evaporation rate [$mm/year$].

500 *Code availability.* The source code of the rSHUD model is kept updating at (https://github.com/SHUD-System/rSHUD, accessed June 2023) and uploaded to Zenodo (Shu, 2023a) . The help files are embedded in the rSHUD R package, user can use *help(FunctionName)* or *?FunctionName* to read the help page. *ls("package:rSHUD")* returns the full list of the functions in the rSHUD Package.

Data availability. The data for building the Shale hills and Waerma Watershed model is embedded in the rSHUD package (Shu, 2023a) (https://github.com/SHUD-System/rSHUD, accessed June 2023). The model output of Waerma Watershed from SHUD model is archived

505 on Zenodo (Shu, 2023b)

Sample availability. No sample is used in this study.

Video supplement. No video is used in this study.

Appendix A: Required R libraries by rSHUD

Table A1. R packages required in rSHUD and their functionalities.

package	Version	Use in rSHUD
Rcpp	1.0	To support the C/C++ program.
reshape2	1.4	Convert any data to data.frame type.
ggplot2	3.4	Powerful plot functions.
gridExtra	2.3	Plot functions.
grid	4.2	Graphics package
fields	10.3	Tools for spatial data
xts	0.10	Tools for time series data.
hydroGOF	0.4	To calculate the goodness of fitting.
zoo	1.8	Tools for time series data.
raster	3.5	Raster data modeling and analysis.
sp	1.5	Vector data modeling and analysis.
rgeos	0.5	Geoanalysis tools.
RTriangle	1.6	Jonathan Shewchuk's Triangle library
rgdal	1.5	Geospatial data library
proj4	1.0	PROJ.4 cartographic projections library
abind	1.0	Combine multidimensional arrays
utils	4.2	R Utils Package
lubridate	1.9	Handle the date-time data.
geometry	0.4	Mesh Generation and Surface Tessellation
methods	4.2	To heritage functions.
ncdf4	1.19	To support the NetCDF data.
GGally	2.1	Extension of ggplot2.
doParallel	1.0	Parallel computing.

Appendix B: Land cover parameter table

510   The table headers in this section have specific meanings as delineated below:

- INDEX: This is the index assigned to each row.

- ALBEDO: This refers to the land cover albedo represented as a dimensionless quantity.

- VEGFRAC: This parameter indicates the vegetation fraction and is expressed as a dimensionless quantity.

- ROUGH: This refers to the Manning roughness assigned to the land cover, expressed in units of $sm^{-1/3}$.

515

- RZD: This is the root depth of the vegetation and is expressed in units of m.

- SOILDGRD: This parameter indicates the soil degradation ratio, given as a dimensionless quantity.

- IMPAF: This parameter indicates the impervious fraction of the land cover, expressed as a dimensionless quantity.

- Classification: This refers to the name of the classification in its original datasets.

B1 UMD land cover classification

Table B1. The parameters for the University of Maryland (UMD) Global Land Cover Classification (Hansen et al., 2000).

INDEX	ALBEDO	VEGFRAC	ROUGH	RZD	SOILDGRD	IMPAF	Classification
0	0.0700	0.0000	0.02	0.0000	0	0	0. Water / Goode's Interrupted Space
1	0.0620	0.8000	0.06	1.0000	0	0	1. Evergreen Needleleaf Forest
2	0.0760	0.9000	0.07	1.2500	0	0	2. Evergreen Broadleaf Forest
3	0.0620	0.8000	0.06	1.0000	0	0	3. Deciduous Needleleaf Forest
4	0.0920	0.8000	0.06	1.2500	0	0	4. Deciduous Broadleaf Forest
5	0.0690	0.7950	0.06	1.1250	0	0	5. Mixed Cover
6	0.0752	0.7999	0.05	0.9975	0	0	6. Woodland
7	0.0908	0.8018	0.04	0.8721	0	0	7. Wooded Grassland
8	0.0991	0.6250	0.05	0.6508	0	0	8. Closed Shrubland
9	0.1213	0.2182	0.04	0.5777	0	0	9. Open Shrubland
10	0.1073	0.7255	0.04	0.7500	0	0	10. Grassland
11	0.1005	0.8354	0.04	0.7500	0.5	0	11. Cropland
12	0.1595	0.0749	0.03	0.5500	0.6	0	12. Bare Ground
13	0.0971	0.7436	0.02	0.7972	0.9	0.9	13. Urban and Built-Up

Table B2. The parameters for USGS 0.5-km MODIS Global Land Cover (Broxton et al., 2014).

INDEX	ALBEDO	VEGFRAC	ROUGH	RZD	SOILDGRD	IMPAF	Classification
0	0.080	0.000	0.020	0.000	0.000	0.000	Water
1	0.140	0.800	0.070	1.0000	0.000	0.000	Evergreen Needle leaf Forest
2	0.100	0.900	0.070	1.2500	0.000	0.000	Evergreen Broadleaf Forest
3	0.140	0.800	0.070	1.0000	0.000	0.000	Deciduous Needle leaf Forest
4	0.120	0.800	0.070	1.2500	0.000	0.000	Deciduous Broadleaf Forest
5	0.110	0.700	0.060	1.1250	0.000	0.000	Mixed Forests
6	0.120	0.700	0.060	0.6508	0.000	0.000	Closed Shrublands
7	0.180	0.500	0.050	0.5777	0.000	0.000	Open Shrublands
8	0.100	0.625	0.045	0.9975	0.000	0.000	Woody Savannas
9	0.150	0.218	0.045	0.8721	0.000	0.000	Savannas
10	0.150	0.726	0.040	0.7500	0.000	0.000	Grasslands
11	0.100	0.200	0.035	0.600	0.000	0.000	Permanent Wetland
12	0.250	0.835	0.040	0.7500	0.500	0.000	Croplands
13	0.246	0.200	0.010	0.7972	0.900	0.900	Urban and Built-Up
14	0.200	0.835	0.040	0.7500	0.500	0.000	Cropland/Natural Vegetation Mosaic
15	0.650	0.000	0.020	0.000	0.000	0.500	Snow and Ice
16	0.300	0.010	0.035	0.5500	0.600	0.000	Barren or Sparsely Vegetated

Table B3. The parameters for China Land Cover Dataset (Yang and Huang, 2021).

INDEX	ALBEDO	VEGFRAC	ROUGH	RZD	SOILDGRD	IMPAF	Classification
1	0.200	0.835	0.040	0.75	0.5	0.0	Cropland
2	0.150	0.800	0.070	1.00	0.0	0.0	Forest
3	0.150	0.600	0.060	0.65	0.0	0.0	Shrub
4	0.150	0.726	0.040	0.75	0.0	0.0	Grassland
5	0.080	0.000	0.020	0.00	0.0	0.0	Water
6	0.650	0.000	0.020	0.00	0.0	0.5	Snow/Ice
7	0.300	0.010	0.035	0.55	0.6	0.0	Barren
8	0.246	0.200	0.010	0.80	0.9	0.9	Impervious
9	0.100	0.200	0.035	0.60	0.0	0.0	Wetland

Appendix C: Pedotransfer function

$$
\begin{aligned}
K_{\text{sat}} =& \exp(7.755 + 0.03252 * p_s + 0.93 * t_p - 0.967 * \rho_b * \rho_b - 0.000484 * p_c * p_c \\
& - 0.000322 * p_s * p_s + 0.001/p_s - 0.0748/p_{\text{om}} - 0.643 * \log(p_s) - 0.01398 * \rho_b * p_c \\
& - 0.1673 * \rho_b * p_{\text{om}} + 0.02986 * t_p * p_c - 0.03305 * t_p * p_s)
\end{aligned}
$$

$$
\begin{aligned}
\theta =& (0.7919 + 0.001691 * p_c - 0.29619 * \rho_b \\
& - 0.000001491 * p_s * p_s + 0.0000821 * p_{\text{om}} * p_{\text{om}} + 0.02427/p_c + = \\
& + 0.01113/p_s + 0.01472 * \log(p_s) - 0.0000733 * p_{\text{om}} * p_c - 0.000619 * \rho_b * p_c \\
& - 0.001183 * \rho_b * p_{\text{om}} - 0.0001664 * t_p * p_s)
\end{aligned}
$$

$$
\begin{aligned}
\alpha =& 100 * \exp(-14.96 + 0.03135 * p_c + 0.0351 * p_s + 0.646 * p_{\text{om}} \\
& + 15.29 * \rho_b - 0.192 * t_p - 4.671 * \rho_b * \rho_b - 0.000781 * p_c * p_c - \\
& - 0.00687 * p_{\text{om}} * p_{\text{om}} + 0.0449/p_{\text{om}} + 0.0663 * \log(p_s) + 0.1482 * \log(p_{\text{om}}) \\
& - 0.04546 * \rho_b * p_s - 0.4852 * \rho_b * p_{\text{om}} + 0.00673 * t_p * p_c)
\end{aligned}
$$

$$
\begin{aligned}
\beta =& 1 + \exp(-25.23 - 0.02195 * p_c + 0.0074 * p_s - 0.1940 * p_{\text{om}} + 45.5 * \rho_b \\
& - 7.24 * \rho_b * \rho_b + 0.0003658 * p_c * p_c + 0.002885 * p_{\text{om}} * p_{\text{om}} - 12.81/\rho_b - 0.1524/p_s - 0.01958/p_{\text{om}} \\
& - 0.2876 * \log(p_s) - 0.0709 * \log(p_{\text{om}}) - 44.6 * \log(\rho_b) - 0.02264 * \rho_b * p_c \\
& + 0.0896 * \rho_b * p_{\text{om}} + 0.00718 * t_p * p_c)
\end{aligned}
$$

Where:

- K_{sat} - Saturated conductivity $[ms^{-1}]$.

- θ - Porosity of soil/geology layer $[m^3 m^{-3}]$.

- α - Coefficient in van Genutchen equation $[m^{-1}]$.

- β - Coefficient in van Genutchen equation $[-]$.

- p_s - Weight percentage of silt in soil $[\%]$.

- p_c - Weight percentage of clay in soil $[\%]$.

- p_{om} - Weight percentage of organic matter in soil $[\%]$.

- ρ_b - Bulk density of soil [%].

- t_p - Flag indicating the top/bottom layer. $t_p = 0$, top layer; $t_p = 1$, bottom layer.

Appendix D: R script

D1 Model deployment, Shale Hill CZO

```
550  rm(list=ls())
     clib=c('rgdal', 'rgeos', 'raster', 'sp', 'fields')
     x=lapply(clib, library, character.only=T)

     library(rSHUD)
555  prjname = 'sh'
     model.in <- file.path('../demo/input', prjname)
     model.out <- file.path('../demo/output', paste0(prjname, '.out'))
     fin=shud.filein(prjname, inpath = model.in, outpath = model.out )

560  dir.create(model.in, showWarnings = F, recursive = T)

     load("./data/sh.rda")
     wbd=sh[['wbd']]
     riv=sh[['riv']]
565  dem=sh[['dem']]
     tsd.forc=sh[['forc']]

     # sl = terrain(dem, v=slope, unit='tangent')
     # cellStats(sl, quantile)
570
     a.max = 200;
     q.min = 33;
     tol.riv = 5
     tol.wb = 5
575  aqd=3
     NX = 800
     years = seq(as.numeric(format(min(time(tsd.forc)), '%Y')),
```

```
                as.numeric(format(max(time(tsd.forc)), '%Y')))
      ndays = days_in_year(years)
580

      riv.simp = rgeos::gSimplify(riv, tol=tol.riv, topologyPreserve = T)
      riv.simp = sp.CutSptialLines(sl=riv.simp, tol=20)

585   wb.dis = rgeos::gUnionCascaded(wbd)
      wb.simp = rgeos::gSimplify(wb.dis, tol=tol.wb, topologyPreserve = T)

      # shp.riv =raster::crop(riv.simp, wb.simp)
      # shp.wb = raster::intersect( wb.simp, riv.simp)
590
      tri = shud.triangle(wb=wb.simp,q=q.min, a=a.max, S=NX)
      # generate   .sp.mesh
      pm=shud.mesh(tri,dem=dem, AqDepth = aqd)
      sp.mesh=sp.mesh2Shape(pm=pm)
595   ncell = nrow(pm@mesh)
      print(ncell)
      # generate   .sp.att

      pa=shud.att(tri)
600
      # generate   .riv
      pr=shud.river(riv.simp, dem = dem)

      # Cut the rivers with triangles
605   spm = sp.mesh2Shape(pm)
      crs(spm) =crs(riv)
      spr=riv
      sp.seg = sp.RiverSeg(spm, spr)
      # Generate the River segments table
610   prs = shud.rivseg(sp.seg)

      # Generate initial condition
```

```
         pic = shud.ic(nrow(pm@mesh), nrow(pr@river), AqD = aqd, p1 = 0.2, p2=0.2)

615  go.plot <- function(){
         z=getElevation(pm = pm)
         loc = getCentroid(pm=pm)
         idx.ord = order(z)
         col=colorspace::diverge_hcl(length(z))
620      plot(sp.mesh[idx.ord, ], axes=TRUE, col=col, lwd=.5); plot(spr , col='blue', add=T, lwd=3)
         # image.plot( legend.only=TRUE, zlim= range(z), col=col, horizontal = T,legend.lab="Elevat
         #               smallplot= c(.6,.9, 0.22,0.26))
         image.plot( legend.only=TRUE, zlim= range(z), col=col, horizontal = F,
                     legend.args = list('text'='Elevation (m)', side=3, line=.05, font=2, adj= .2),
625                  smallplot= c(.79,.86, 0.20,0.4))
     }
     ia = getArea(pm=pm)
     png(filename = '~/sh_mesh.png', height = 9, width = 6, res = 400, units = 'in')
     par(mfrow=c(2,1), mar=c(3, 3.5, 1.5,1) )
630  go.plot();
     mtext(side=3, text = '(a)')
     hist(ia, xlab='', nclass=20, main='', ylab='')
     mtext(side=3, text = '(b)')
     mtext(side=2, text = 'Frequency', line=2)
635  mtext(side=1, text = expression(paste("Area (", m^2, ")")), line=2)
     box();
     # grid()
     dev.off()

640
     sp.c = SpatialPointsDataFrame(gCentroid(wb.simp, byid = TRUE),
                                   data=data.frame('ID' = 'forcing'), match.ID = FALSE)
     sp.forc = ForcingCoverage(sp.meteoSite = sp.c,  pcs=crs(wb.simp), dem=dem, wbd=wbd)
     write.forc(sp.forc@data,
645           path = file.path('./input', prjname),
              startdate = format(min(time(tsd.forc)), '%Y%m%d'),
              file=fin['md.forc'])
```

```
      write.tsd(tsd.forc, file = file.path(fin['inpath'], 'forcing.csv'))

650   # model configuration, parameter
      cfg.para = shud.para(nday=ndays)
      # calibration
      cfg.calib = shud.calib()

655   para.lc = lc.NLCD(lc=42) # 42 is the forest in NLCD classes
      para.soil = PTF.soil()
      para.geol = PTF.geol()

      tsd.lai =  LaiRf.NLCD(lc=42, years=years)
660   write.tsd(tsd.lai$LAI, file = fin['md.lai'])

      tsd.mf = MeltFactor(years=years)
      write.tsd(tsd.mf, file = fin['md.mf'])
      # write  input files.
665   write.mesh(pm, file = fin['md.mesh'])
      write.riv(pr, file=fin['md.riv'])
      write.ic(pic, file=fin['md.ic'])

      write.df(pa, file=fin['md.att'])
670   write.df(prs, file=fin['md.rivseg'])
      write.config(cfg.para, fin['md.para'])
      write.config(cfg.calib, fin['md.calib'])

      write.df(para.lc, file=fin['md.lc'])
675   write.df(para.soil, file=fin['md.soil'])
      write.df(para.geol, file=fin['md.geol'])
      writeshape(riv.simp, file=file.path(dirname(fin['md.att']), 'riv'))
      print(nrow(pm@mesh))
```

D2   Model deployment, Waerma watershed

```
680   rm(list=ls())
      # === 1. load library ============
```

```
     clib=c('rgdal', 'rgeos', 'raster', 'sp', 'fields', 'xts')
     x=lapply(clib, library, character.only=T)
     library(rSHUD)
685
     # === 2. create directories  ============
     dir.prj = '~/Documents/Ex_waerma'
     dir.forc = file.path(dir.prj, 'forc')
     dir.fig = file.path(dir.prj, 'figure')
690  dir.create(dir.forc, showWarnings = FALSE, recursive = TRUE)
     dir.create(dir.fig, showWarnings = FALSE, recursive = TRUE)

     # === 3. setup the project ============
     prjname = 'waerma'
695  model.in <- file.path(dir.prj, 'input', prjname)
     model.out <- file.path(dir.prj, 'output', paste0(prjname, '.out'))
     fin=shud.filein(prjname, inpath = model.in, outpath = model.out )
     if (dir.exists(model.in)){
       unlink(model.in, recursive = T, force = T)
700  }
     dir.create(model.in, showWarnings = F, recursive = T)

     # === 4. load and reproject data ============
     data(waerma)
705  wbd=waerma[['wbd']]
     meteosite = waerma[['meteosite']] # This is in GCS

     crs.pcs = crs.Albers(wbd)
     dem = projectRaster(waerma[['dem']], crs=crs.pcs)
710  wbd= spTransform(waerma[['wbd']], CRSobj = crs.pcs)
     riv= spTransform(waerma[['riv']], CRSobj = crs.pcs)

     # sl=mask(terra::terrain(dem, opt='slope', unit='tangent'), wbd)
     # plot(sl)
715
     r0.soil = waerma[['soil']]
```

```
    att.soil = waerma[['att']]$soil
    rcl.soil=cbind(att.soil[, 1], 1:nrow(att.soil))
    r.soil = projectRaster(reclassify(r0.soil, rcl.soil), crs = crs.pcs, method="ngb")
720
    r0.geol = waerma[['geol']]
    att.geol = waerma[['att']]$geol
    rcl.geol=cbind(att.geol[, 1], 1:nrow(att.geol))
    r.geol = projectRaster(reclassify(r0.geol, rcl.geol), crs = crs.pcs, method="ngb")
725
    r0.lc = waerma[['lc']]
    att.lc = waerma[['att']]$lc
    rcl.lc=cbind(att.lc[, 1], 1:nrow(att.lc))
    r.lc = projectRaster(reclassify(r0.lc, rcl.lc), crs = crs.pcs, method="ngb")
730
    # === 5. some threshold for model deployment ============
    AREA = 9853260 # KNOWN Area
    a.max = 150*150;
    q.min = 33;
735 tol.riv = 50
    tol.wb = 50
    aqd = 6
    NX = AREA / a.max
    ndays = 731
740
    # === 6. domain decomposition ============
    # simplify the river network.
    riv.simp = rgeos::gSimplify(riv, tol=tol.riv, topologyPreserve = T)

745 # desolve and simplify the watershed boundary
    wb.dis = rgeos::gUnionCascaded(wbd)
    wb.simp = rgeos::gSimplify(wb.dis, tol=tol.wb, topologyPreserve = T)

    # !! Triangulation
750 tri = shud.triangle(wb=wb.simp,q=q.min, a=a.max, S=NX)
    # generate  .sp.mesh
```

```r
    pm=shud.mesh(tri,dem=dem, AqDepth = aqd)
    sp.mesh=sp.mesh2Shape(pm=pm)
    ncell = nrow(pm@mesh)
755 print(ncell)

    # generate  .riv
    pr=shud.river(riv.simp, dem = dem)

760
    # === 7. TSD DATA ============
    fns.meteo  =  paste0(meteosite$FILENAME, '.csv')
    tsd.forc = waerma$tsd$forc
    range(time(tsd.forc[[1]]))
765 for(i in 1:length(fns.meteo)){
      write.tsd(tsd.forc[[i]], file = file.path(dir.forc, fns.meteo[i]))
    }
    tsd.mf = MeltFactor(years = seq(as.numeric(format(min(time(tsd.forc[[1]])), '%Y')),
                                    as.numeric(format(max(time(tsd.forc[[1]])), '%Y'))))
770 tsd.lai = waerma$tsd$lai[[1]]
    # Coverage of meteorological sites.
    sp.forc = ForcingCoverage(sp.meteoSite = meteosite,
                                 filenames= fns.meteo,
                                 pcs=crs.pcs, gcs=crs(meteosite),
775                              dem=dem, wbd=wbd)
    write.forc(sp.forc@data, path = normalizePath(dir.forc),
               startdate = format(min(time(tsd.forc[[1]])), '%Y%m%d'),
               file=fin['md.forc'])

780 # === 8. attributes ============
    # generate  .sp.att
    pa=shud.att(tri, r.soil = r.soil, r.geol = r.geol, r.lc=r.lc, r.forc = sp.forc)
    head(pa)

785 # === 9. toplogical relation between river and triangle  ============
    # Cut the rivers with triangles
```

```
      spm = sp.mesh2Shape(pm)
      crs(spm) =crs(riv)
      spr=riv
790   sp.seg = sp.RiverSeg(spm, spr)
      # Generate the River segments table
      prs = shud.rivseg(sp.seg)

      # Generate initial condition
795   pic = shud.ic(nrow(pm@mesh), nrow(pr@river), AqD = aqd)

      go.plot <- function(){
        z=getElevation(pm = pm)
        loc = getCentroid(pm=pm)
800     idx.ord = order(z)
        col=colorspace::diverge_hcl(length(z))
        plot(sp.mesh[idx.ord, ], axes=TRUE, col=col, lwd=.5); plot(spr , col='blue', add=T, lwd=3)
        image.plot( legend.only=TRUE, zlim= range(z), col=col, horizontal = F,
                    legend.args = list('text'='Elevation (m)', side=3, line=.05, font=2, adj= .2),
805                 smallplot= c(.79,.86, 0.20,0.4))
      }
      ia = getArea(pm=pm)
      png(filename = file.path(dir.fig, paste0(prjname, '_mesh.png')), height = 9, width = 6, res
      par(mfrow=c(2,1), mar=c(3, 3.5, 1.5,1) )
810   go.plot();
      mtext(side=3, text = '(a)')
      mtext(side=3, text = paste0('Ncell = ', ncell), line=-1)
      hist(ia, xlab='', nclass=20, main='', ylab='')
      mtext(side=3, text = '(b)')
815   mtext(side=2, text = 'Frequency', line=2)
      mtext(side=1, text = expression(paste("Area (", m^2, ")")), line=2)
      box();
      # grid()
      dev.off()
820
      # === 10. configuration files  ============
```

```
     # model configuration, parameter
     cfg.para = shud.para(nday=ndays)

825  # calibration file
     cfg.calib = shud.calib()

     para.lc = lc.GLC()
     para.soil = PTF.soil(att.soil[, -1])  # only 4 columns (Silt, clay, OM, bulk density) as inp
830  para.geol = PTF.geol(att.geol[, -1])

     # === 11. write  input files. ============
     write.mesh(pm, file = fin['md.mesh'])
     write.riv(pr, file = fin['md.riv'])
835  write.ic(pic, file = fin['md.ic'])

     write.df(pa, file=fin['md.att'])
     write.df(prs, file=fin['md.rivseg'])
     write.config(cfg.para, fin['md.para'])
840  write.config(cfg.calib, fin['md.calib'])

     write.tsd(tsd.lai, fin['md.lai'] )
     write.tsd(tsd.mf, fin['md.mf'] )

845  write.df(para.lc, file=fin['md.lc'])
     write.df(para.soil, file=fin['md.soil'])
     write.df(para.geol, file=fin['md.geol'])
     writeshape(riv.simp, file=file.path(dirname(fin['md.att']), 'riv'))
     print(nrow(pm@mesh))
```

850 **D3   Post-processing, Waerma watershed**

```
     rm(list=ls())

     # === pre1. load library ============
     clib=c('rgdal', 'rgeos', 'raster', 'sp', 'fields', 'xts', 'ggplot2')
855  x=lapply(clib, library, character.only=T)
```

```r
    library(rSHUD)

    # === pre2. create directories  ============
    dir.prj = '~/Documents/Ex_waerma'
860 dir.forc = file.path(dir.prj, 'forc')
    dir.fig = file.path(dir.prj, 'figure')
    dir.create(dir.forc, showWarnings = FALSE, recursive = TRUE)
    dir.create(dir.fig, showWarnings = FALSE, recursive = TRUE)

865 # === pre3. setup the project ============
    prjname = 'waerma'
    model.in <- file.path(dir.prj, 'input', prjname)
    # model.out <- file.path(dir.prj, 'output', paste0(prjname, '.out'))
    model.out <- '~/Documents/output/waerma.out'
870
    fin=shud.filein(prjname, inpath = model.in, outpath = model.out )
    shud.env(prjname, inpath = model.in, outpath = model.out )
    dir.create(model.in, showWarnings = F, recursive = T)

875 ia=getArea()
    ncell=length(ia)
    spm=sp.mesh2Shape()

    gplotfun <- function(r, leg.lab='value'){
880   map.p <- rasterToPoints(r)
      #Make the points a dataframe for ggplot
      df <- data.frame(map.p)
      #Make appropriate column headings
      colnames(df) <- c('X', 'Y', 'Value')
885   #Now make the map
      g= ggplot(data=df, aes(y=Y, x=X)) +
        geom_raster(aes(fill=Value)) +
        # geom_point(data=sites, aes(x=x, y=y), color="white", size=3, shape=4) +
        theme_bw() + coord_equal() +
890     # scale_fill_continuous(leg.lab) +
```

```
      theme(
        # axis.title.x = element_text(size=16),
        # axis.title.y = element_text(size=16, angle=90),
        # axis.text.x = element_text(size=14),
895     # axis.text.y = element_text(size=14),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
900     panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = 'right',
        legend.key = element_blank()
      )
905   return(g)
    }
    gl=list()
    # === 1. plot Q (discharge) data ============
    oid = getOutlets()
910 qdown = readout('rivqdown')
    prcp = readout('elevprcp')
    xt = 1:(365*2)+365*1
    q=qdown[xt, oid]
    pq = cbind(q, rowMeans(prcp[xt,]))[,2:1]
915 # gl[[1]] = autoplot(q)+xlab('')+ylab('Discharge (m^3/day)')+theme_bw()
    gl[[1]] = hydrograph(pq, ylabs = c('Preciptation (mm)', 'Discharge (cmd)'))
    gl[[1]]

    # === 2. plot Water Balance ============
920 xl=loaddata(varname=c('rivqdown', 'eleveta', 'elevetp', 'elevprcp', 'eleygw'))
    wb=wb.all(xl=xl, plot=F)[(1:24)+12, ]*1000
    gl[[2]] = hydrograph(wb, ylabs = c('Storage (mm)', 'Flux (mm/mon)'), legend.position='top')
    gl[[2]]

925 # === 3. plot groundwater data ============
```

```r
    eleygw = readout('eleygw')[xt, ]
    ts.gw=apply.daily(eleygw, sum)/ncell
    # plot(ts.gw)
    gw.mean = apply(eleygw, 2, mean)
930 aqd =getAquiferDepth()
    r.gw = MeshData2Raster(gw.mean)
    d.gw = aqd - r.gw
    d.gw[d.gw<0]=0
    gl[[3]] =gplotfun(d.gw, leg.lab='Depth (m)')+
935   scale_fill_gradient(low = "darkblue", high = "yellow")
    gl[[3]]

    # === 4. plot ETa data ============
    eleveta = readout('eleveta')[xt, ]
940 ts.eta=apply.monthly(eleveta, sum)
    # plot(ts.eta)
    eta.mean = apply(eleveta, 2, mean)*365
    r.eta = MeshData2Raster(eta.mean)*1000 # mm/day
    # plot(spm, axes=TRUE)
945 # plot(add=T, r.eta)
    gl[[4]]=gplotfun(r.eta, leg.lab='Rate (mm/day) ')+
      scale_fill_gradient(low = "white", high = "blue")
    gl[[4]]

950 # === Saving the plots ============
    gg=gridExtra::arrangeGrob(grobs=gl, nrow=2, ncol=2)
    ggsave(plot = gg, filename = file.path(dir.fig, 'waerma_res.png'), width = 7, height=7, dpi=

    for(i in 1:4){
955   ggsave(plot = gl[[i]], filename = file.path(dir.fig, paste0('waerma_res_', i, '.png')),
             width = 3.5, height=4, dpi=400, units = 'in')
    }
```
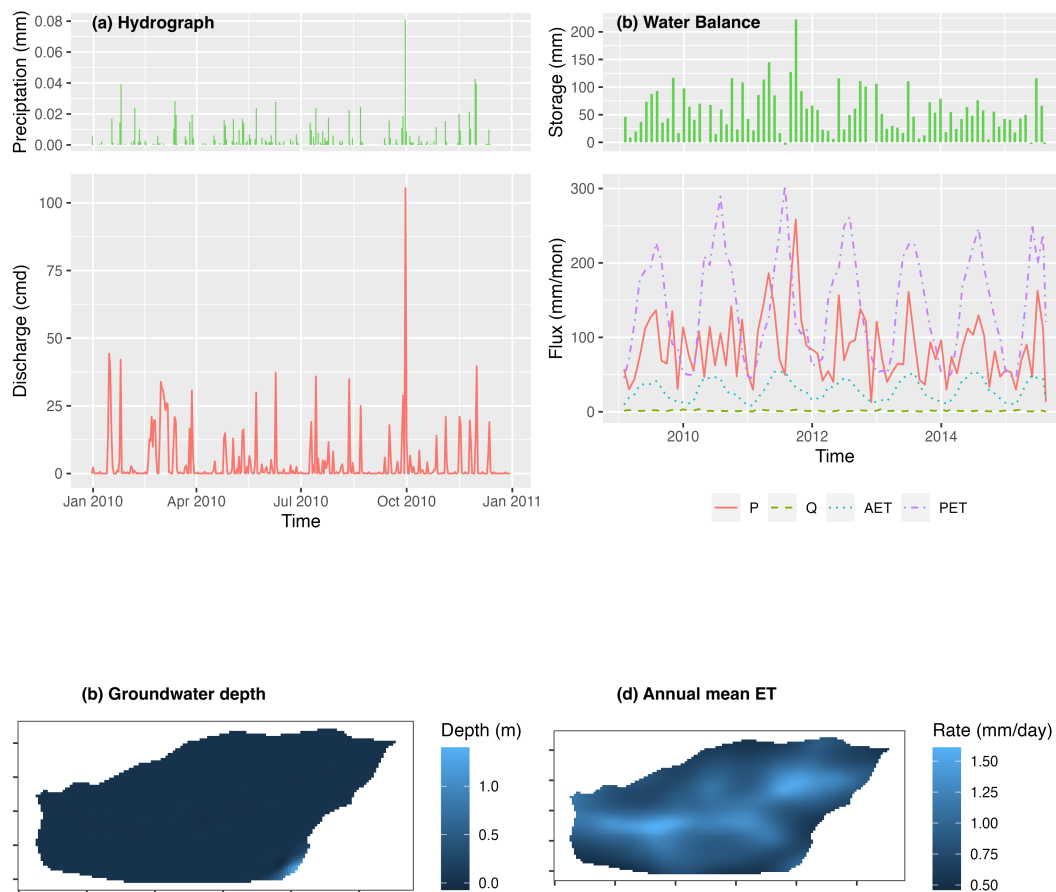
Figure E1. The result analysis of modeling Shale Hills Critical Zone Observatory watershed; (a) hydrograph, (b) watershed scale water balance, (c) ground water depth [m], and (d) annual evaporation rate [$mm/year$].

Appendix E: Results visualization of SHCZO

Author contributions. Lele Shu – Conceptualization, Investigation, Methodology, Software, Validation, Visualization, Writing original draft and editing

Paul Ullrich – Supervision, Investigation, Writing original draft and editing

Xianhong Meng – Supervision, Investigation, Writing original draft and editing

Christopher Duffy – Supervision, Investigation, Writing original draft and editing

Hao Chen – Code development, program test, writing and editing.

965   Zhaoguo Li– Code development, program test, writing and editing.

Competing interests.   Lele Shu and Paul Ullrich are members of the editorial board of the journal

Disclaimer.   TEXT

Acknowledgements.   Authors Shu was supported by the Chinese Academy Sciences (E0290304, xbzg-zdsys-202215), National Cryosphere Desert Data Center (E01Z790215), the Qinghai Key Laboratory of Disaster Prevention (QFZ-2021-Z02) and the California Energy Commis-

970   sion grant (award no. EPC-16-063).

Co-author Meng was supported by the National Natural Science Foundation of China (41930759).

Co-author Ullrich was supported by the U.S. Department of Energy Regional and Global Climate Modeling Program (RGCM) "An Integrated Evaluation of the Simulated Hydroclimate System of the Continental US" project (award no. DE-SC0016605) and the National Institute of Food and Agriculture, U.S. Department of Agriculture, California Agricultural Experiment Station hatch project accession no.

975   1016611.

References

Arnold, J. G., Srinivasan, R., and Muttiah, R. S.: Large area hydrologic modeling and assessment part I: model development, Journal of the American Water Resources Association, 34, 73–89, 1998.

Beven, K.: Rainfall-Runoff Modelling, Wiley, Chichester, UK, https://doi.org/10.1002/9781119951001, 2012.

Beven, K.: So how much of your error is epistemic? Lessons from Japan and Italy, Hydrological Processes, 27, 1677–1680, https://doi.org/10.1002/hyp.9648, 2013.

Beven, K.: Towards a methodology for testing models as hypotheses in the inexact sciences, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 475, https://doi.org/10.1098/rspa.2018.0862, 2019.

Beven, K.: Deep learning, hydrological processes and the uniqueness of place, Hydrological Processes, 34, 3608–3613, https://doi.org/10.1002/hyp.13805, 2020.

Beven, K. and Young, P.: A guide to good practice in modeling semantics for authors and referees, Water Resources Research, 49, 5092–5098, https://doi.org/10.1002/wrcr.20393, 2013.

Beven, K. J.: On hypothesis testing in hydrology: Why falsification of models is still a really good idea, WIREs Water, 5, 1–8, https://doi.org/10.1002/wat2.1278, 2018.

Bhatt, G.: A distributed hydrologic modeling system: Framework for discovery and management of water resources, Ph.D. thesis, Pennsylvania State University, 2012.

Bhatt, G., Kumar, M., and Duffy, C. J.: A tightly coupled GIS and distributed hydrologic modeling framework, Environmental Modelling and Software, 62, 70–84, https://doi.org/10.1016/j.envsoft.2014.08.003, 2014.

Brantley, S. L., White, T., West, N., Williams, J. Z., Forsythe, B., Shapich, D., Kaye, J., Lin, H., Shi, Y., Kaye, M., Herndon, E., Davis, K. J., He, Y., Eissenstat, D., Weitzman, J., DiBiase, R., Li, L., Reed, W., Brubaker, K., and Gu, X.: Susquehanna Shale Hills Critical Zone Observatory: Shale Hills in the Context of Shaver's Creek Watershed, Vadose Zone Journal, 17, 1–19, https://doi.org/10.2136/vzj2018.04.0092, 2018.

Broxton, P. D., Zeng, X., Sulla-Menashe, D., and Troch, P. A.: A Global Land Cover Climatology Using MODIS Data, Journal of Applied Meteorology and Climatology, 53, 1593–1605, https://doi.org/10.1175/JAMC-D-13-0270.1, 2014.

Burnash, R., Ferral, R., McGuire, R., and Center, J. F.-S. R. F.: A Generalized Streamflow Simulation System: Conceptual Modeling for Digital Computers, U. S. Department of Commerce, National Weather Service, and State of California, Department of Water Resources, 1973.

Downing, J. A., Cole, J. J., Duarte, C. M., Middelburg, J. J., Melack, J. M., Prairie, Y. T., Kortelainen, P., Striegl, R. G., McDowell, W. H., and Tranvik, L. J.: Global abundance and size distribution of streams and rivers, Inland Waters, 2, 229–236, https://doi.org/10.5268/IW-2.4.502, 2012.

Duffy, C. J.: The terrestrial hydrologic cycle: an historical sense of balance, WIREs Water, 4, e1216, https://doi.org/10.1002/wat2.1216, 2017.

Flint, J. J.: Stream gradient as a function of order, magnitude, and discharge, Water Resources Research, 10, 969–973, https://doi.org/10.1029/WR010i005p00969, 1974.

Hansen, M. C., Sohlberg, R., Defries, R. S., and Townshend, J. R.: Global land cover classification at 1 km spatial resolution using a classification tree approach, vol. 21, https://doi.org/10.1080/014311600210209, 2000.

He, J., Yang, K., Tang, W., Lu, H., Qin, J., Chen, Y., and Li, X.: The first high-resolution meteorological forcing dataset for land process studies over China, Scientific Data, 7, 25, https://doi.org/10.1038/s41597-020-0369-y, 2020.

Hock, R.: Temperature index melt modelling in mountain areas, Journal of Hydrology, 282, 104–115, https://doi.org/10.1016/S0022-1694(03)00257-9, 2003.

Jin, L., Andrews, D. M., Holmes, G. H., Lin, H., and Brantley, S. L.: Opening the "Black Box": Water Chemistry Reveals Hydrological Controls on Weathering in the Susquehanna Shale Hills Critical Zone Observatory, Vadose Zone Journal, 10, 928–942, https://doi.org/10.2136/vzj2010.0133, 2011.

Kratzer, J. F., Hayes, D. B., and Thompson, B. E.: Methods for interpolating stream width, depth, and current velocity, Tech. Rep. 1-2, https://doi.org/10.1016/j.ecolmodel.2006.02.004, 2006.

Kumar, S., Sekhar, M., Reddy, D. V., and Mohan Kumar, M. S.: Estimation of soil hydraulic properties and their uncertainty: Comparison between laboratory and field experiment, Hydrological Processes, 24, 3426–3435, https://doi.org/10.1002/hyp.7775, 2010.

Li, X., Khandelwal, A., Jia, X., Cutler, K., Ghosh, R., Renganathan, A., Xu, S., Tayal, K., Nieber, J., Duffy, C., Steinbach, M., and Kumar, V.: Regionalization in a Global Hydrologic Deep Learning Model: From Physical Descriptors to Random Vectors, Water Resources Research, 58, https://doi.org/10.1029/2021WR031794, 2022.

Lin, H.: Temporal Stability of Soil Moisture Spatial Pattern and Subsurface Preferential Flow Pathways in the Shale Hills Catchment, Vadose Zone Journal, 5, 317–340, https://doi.org/10.2136/vzj2005.0058, 2006.

McManamay, R. A. and DeRolph, C. R.: A stream classification system for the conterminous United States, Scientific Data, 6, 190 017, https://doi.org/10.1038/sdata.2019.17, 2019.

Meng, X., Lyu, S., Li, Z., Ao, Y., Wen, L., Shang, L., Wang, S., Deng, M., Zhang, S., Zhao, L., Chen, H., Ma, D., Li, S., Shu, L., An, Y., and Niu, H.: Dataset of Comparative Observations for Land Surface Processes over the Semi-Arid Alpine Grassland against Alpine Lakes in the Source Region of the Yellow River, Advances in Atmospheric Sciences, 40, 1142–1157, https://doi.org/10.1007/s00376-022-2118-y, 2023.

Nachtergaele, F., van Velthuizen, H., Verelst, L., Batjes, N., Dijkshoorn, J., van Engelen, V., Fischer, G., Jones, A., Montanarella, L., Petri, M., Prieler, S., Teixeira, E., Wilberg, D., and Shi, X.: Harmonized World Soil Database (version 1.0), Food and Agric Organization of the UN (FAO); International Inst. for Applied Systems Analysis (IIASA); ISRIC - World Soil Information; Inst of Soil Science-Chinese Acad of Sciences (ISS-CAS); EC-Joint Research Centre (JRC), 2008.

NASA, METI, AIST, Japan Spacesystems, and U.S./Japan ASTER Science Team: ASTER Global Digital Elevation Model V003, https://doi.org/10.5067/ASTER/ASTGTM.003, 2018.

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., and Gupta, H. V.: What Role Does Hydrological Science Play in the Age of Machine Learning?, Water Resources Research, 57, https://doi.org/10.1029/2020WR028091, 2021.

Newman, A. J., Mizukami, N., Clark, M. P., Wood, A. W., Nijssen, B., and Nearing, G.: Benchmarking of a physically based hydrologic model, Journal of Hydrometeorology, 18, 2215–2225, https://doi.org/10.1175/JHM-D-16-0284.1, 2017.

Paniconi, C. and Putti, M.: Physically based modeling in catchment hydrology at 50: Survey and outlook, Water Resources Research, 51, 7090–7129, https://doi.org/10.1002/2015WR017780, 2015.

Peckham, S. D., Stoica, M., Jafarov, E., Endalamaw, A., and Bolton, W. R.: Reproducible, component-based modeling with TopoFlow, a spatial hydrologic modeling toolkit, Earth and Space Science, 4, 377–394, https://doi.org/10.1002/2016EA000237, 2017.

Peel, M. C. and McMahon, T. A.: Historical development of rainfall-runoff modeling, WIREs Water, 7, 1–15, https://doi.org/10.1002/wat2.1471, 2020.

Perron, J. T. and Royden, L.: Earth Surface Processes and Landforms, 38, 570–576, https://doi.org/10.1002/esp.3302, 2013.

Qu, Y. and Duffy, C. J.: A semidiscrete finite volume formulation for multiprocess watershed simulation, Water Resources Research, 43, 1–18, https://doi.org/10.1029/2006WR005752, 2007.

Shewchuk, J. R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in: Applied Computational Geometry: Towards Geometric Engineering, edited by Lin, M. C. and Manocha, D., vol. 1148 of *Lecture Notes in Computer Science*, pp. 203–222, Springer-Verlag, from the First ACM Workshop on Applied Computational Geometry, 1996.

Shewchuk, J. R.: Delaunay refinement algorithms for triangular mesh generation, Computational Geometry, 22, 21–74, https://doi.org/https://doi.org/10.1016/S0925-7721(01)00047-5, 16th ACM Symposium on Computational Geometry, 2002.

Shi, Y., Davis, K. J., Duffy, C. J., and Yu, X.: Development of a Coupled Land Surface Hydrologic Model and Evaluation at a Critical Zone Observatory, Journal of Hydrometeorology, 14, 1401–1420, https://doi.org/10.1175/JHM-D-12-0145.1, 2013.

Shu, L.: SHUD-System/SHUD_User_Guide: v1.0, https://doi.org/10.5281/zenodo.3561295, 2019.

Shu, L.: SHUD-System/rSHUD: 2.0, https://doi.org/10.5281/zenodo.8104336, 2023a.

Shu, L.: Output from SHUD hydrological model for Waerma Watershed, Gansu, China, https://doi.org/10.5281/zenodo.8104324, 2023b.

Shu, L., Ullrich, P. A., and Duffy, C. J.: Simulator for Hydrologic Unstructured Domains (SHUD v1.0): numerical modeling of watershed hydrology with the finite volume method, Geoscientific Model Development, 13, 2743–2762, https://doi.org/10.5194/gmd-13-2743-2020, 2020.

Shu, L. e. a.: Global Hydrologic Data Cloud: An Open, Quick and Reproducible Methodology for Watershed Modeling (Submitted), Journal of Hydrology, https://doi.org/xxxx, 2023c.

Strahler, A. N.: Hypsometric (area-altitude) analysis of erosional topography, Bulletin of the Geological Society of America, 63, 1117–1142, 1952.

Strick, R. J., Ashworth, P. J., Awcock, G., and Lewin, J.: Morphology and spacing of river meander scrolls, Geomorphology, 310, 57–68, https://doi.org/10.1016/j.geomorph.2018.03.005, 2018.

Wickham, J. D., Homer, C., Fry, J., Boyte, K. G., Yang, L., Bright, B., and Coulston, J.: Conterminous United States land cover change patterns 2001–2016 from the 2016 National Land Cover Database, International Journal of Applied Earth Observation and Geoinformation, 91, 102 154, https://doi.org/https://doi.org/10.1016/j.jag.2020.102154, 2020.

Wösten, J. H. M., Pachepsky, Y. a., and Rawls, W. J.: Pedotransfer functions: Bridging the gap between available basic soil data and missing soil hydraulic characteristics, Journal of Hydrology, 251, 123–150, https://doi.org/10.1016/S0022-1694(01)00464-4, 2001.

Yang, J. and Huang, X.: The 30 m annual land cover dataset and its dynamics in China from 1990 to 2019, Earth System Science Data, 13, 3907–3925, https://doi.org/10.5194/essd-13-3907-2021, 2021.

Yu, X., Duffy, C., Baldwin, D. C., and Lin, H.: The role of macropores and multi-resolution soil survey datasets for distributed surface–subsurface flow modeling, Journal of Hydrology, 516, 97–106, https://doi.org/10.1016/j.jhydrol.2014.02.055, 2014.

Zhang, S., Ye, B., Liu, S., Zhang, X., and Hagemann, S.: A modified monthly degree-day model for evaluating glacier runoff changes in China. Part I: model development, Hydrological Processes, 26, 1686–1696, https://doi.org/10.1002/hyp.8286, 2012.