Technical Note

# HydroCNHS: A Python Package of Hydrological Model for Coupled Natural–Human Systems

Chung-Yi Lin, S.M.ASCE[1]; Yi-Chen Ethan Yang, Ph.D., A.M.ASCE[2]; and Sungwook Wi, Ph.D., A.M.ASCE[3]

**Abstract:** Modeling coupled natural–human systems (CNHS) to inform comprehensive water resources management policies or describe hydrological cycles in the Anthropocene has become popular in recent years. To fulfill this need, we developed a semidistributed hydrological model for coupled natural–human systems, HydroCNHS. HydroCNHS is an open-source Python package supporting four application programming interfaces (APIs) that enable users to integrate their human decision models, which can be programmed with the agent-based modeling concept, into HydroCNHS. Specifically, we designed Dam API, RiverDiv API, Conveying API, and InSitu API to integrate, respectively, customized man-made infrastructures such as reservoirs, off-stream diversions, transbasin aqueducts, and drainage systems that abstract human behaviors (e.g., operator and farmer water use decisions). Each of the HydroCNHS APIs has a unique plug-in structure that respects within-subbasin and inter-subbasin (i.e., river) routing logic for maintaining the water balance. In addition, HydroCNHS uses a single model configuration file to organize input features for the hydrological model and case-specific human systems models. Also, HydroCNHS enables model calibration using parallel computing power. We demonstrate the functionalities of the HydroCNHS package through a case study in the Northwest United States. Given the integrity of the modeling framework, HydroCNHS can benefit water resources planning and management in various aspects, including uncertainty analysis in CNHS modeling and more complex agent design.
**DOI: 10.1061/(ASCE)WR.1943-5452.0001630.** 

**Author keywords:** Open source; Agent-based modeling (ABM); Human decision; Human behavior; Transboundary analysis.



## Introduction

Recently, many studies have explored the coevolution of natural and human water systems with coupled natural–human systems (CNHS) modeling approach, e.g., Faust et al. (2017) and Wada et al. (2017), for a comprehensive evaluation of water resources management policies (Yang et al. 2020) and near-surface water cycles (Sivapalan and Blöschl 2015). The coupled modeling approach often consists of a process-based hydrological model and a human infrastructure model. Agent-based modeling (ABM) is commonly adopted to describe heterogeneous human behaviors and their impacts on water systems that significantly vary at various spatial and temporal scales, e.g., Hu and Beattie (2019), Lin et al. (2022), and Lin and Yang (2022). Each agent represents a decision-making unit defined by a set of attributes and behavior rules. In general, human-made infrastructures such as reservoirs, diversions, transbasin aqueducts, and drainage systems can be represented as an agent and coupled with hydrological models with desired bidirectional information exchange frequency.

However, developing a sophisticated human model is not always possible for CNHS modeling/modelers owing to the lack of data or other limitations. For example, when modeling reservoir releases, modelers can use historical records (e.g., daily time series) as exogenous inputs or use a decision-making model to endogenously and dynamically simulate water releases. While some existing hydrological model software, e.g., Arnold et al. (2012) and Liang et al. (1996), can incorporate human decision units, the option that allows users to choose among exogenous or endogenous human components is often not supported. Knox et al. (2018) developed a generic network-based multiagent framework to link natural models and human models, which is one of the earlier efforts to address this gap. Following Knox et al. (2018) and trying to specifically target the water system, this technical note aims to develop a semidistributed hydrological model for coupled natural–human systems (hereafter HydroCNHS) that facilitates integrating hydrologic models with agent-based human system models through a generalizable coupling procedure with four application programming interfaces (APIs). The four APIs are Dam API, RiverDiv API, Conveying API, and InSitu API, which have distinct plug-in structures that respect within-subbasin and inter-subbasin (i.e., river) routing logic for maintaining the water balance. They can integrate human models, where heterogeneous human agents can be modeled with different decision-making process complexity and data intensity (e.g., exogenous input data or endogenous rules) from a bottom-up viewpoint. Essentially, HydroCNHS is a Python package simulating natural and human-induced water cycles within one or multiple watershed systems on a daily scale. The package features a single model configuration file to organize input settings for hydrological models and case-specific human models. In addition, HydroCNHS supports a parallel calibration module using a genetic algorithm (GA; Whitley 1994). The package is published with a GPL-3.0 License to follow the concept of Open Science (NASEM 2021). We demonstrate the functionalities of HydroCNHS in a case

[1]Graduate Research Assistant, Dept. of Civil and Environmental Engineering, Lehigh Univ., Bethlehem, PA 18015. ORCID: https://orcid.org/0000-0002-9464-4696. Email: chl319@lehigh.edu

[2]Assistant Professor, Dept. of Civil and Environmental Engineering, Lehigh Univ., Bethlehem, PA 18015 (corresponding author). ORCID: https://orcid.org/0000-0002-7982-7988. Email: yey217@lehigh.edu

[3]Research Assistant Professor, Dept. of Civil and Environmental Engineering, Univ. of Massachusetts, Amherst, MA 01003. Email: sungwookwi@gmail.com

Note. This manuscript was submitted on February 13, 2022; approved on August 27, 2022; published online on October 13, 2022. Discussion period open until March 13, 2023; separate discussions must be submitted for individual papers. This technical note is part of the ***Journal of Water Resources Planning and Management***, © ASCE, ISSN 0733-9496.

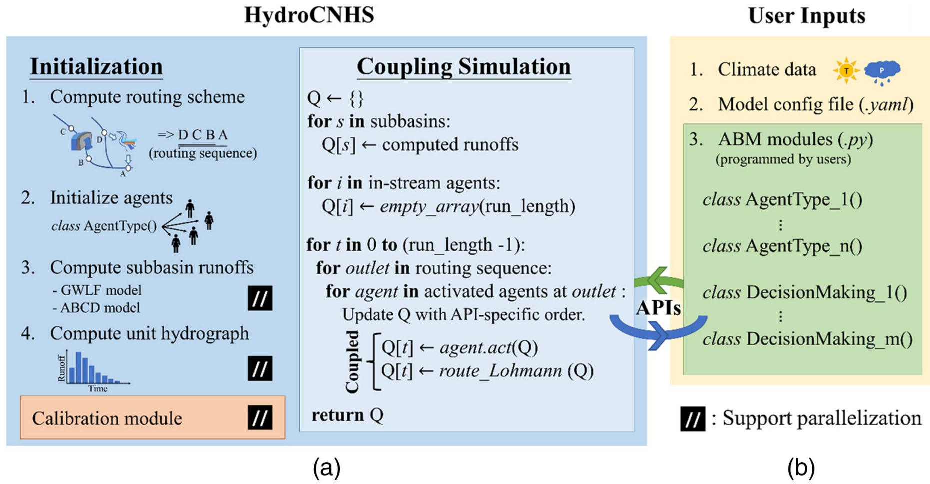


**Fig. 1.** (a) HydroCNHS model structure; and (b) three user inputs, including climate data (temperature and precipitation), model configuration file (*.yaml*), and ABM modules (*·py*; green box). User-provided ABM modules will be integrated into HydroCNHS through four APIs.

study with the Tualatin River Basin (TRB) in the Northwest United States.

## Methods

### Structure of the HydroCNHS Model

The HydroCNHS Python package is a semidistributed hydrological model for CNHS that simulates natural and human-induced water cycles on a daily scale. The subbasin delineation is based on the agent design and user-desired distributing resolution. Fig. 1 shows the HydroCNHS model structure (blue box) and user inputs (yellow box). Three inputs are required: (1) daily climate data (precipitation and temperature), (2) a model configuration file (*.yaml*; setting for the HydroCNHS and ABM modules), and (3) ABM modules (*.py*; green box). HydroCNHS APIs handle the logic to integrate ABM modules.

In the "Initialization" step (Fig. 1), HydroCNHS forms the routing scheme based on the stream orders associated with outlets (i.e., routing order of outlets). Then, agent instances/objects are created according to user-defined agent classes in ABM modules (*.py*). A "class" is a data structure in object-oriented Python defined by "attributes" and "methods." Once initialized, each agent is an instance of an assigned agent class. For example, two reservoir agents can be created by a single reservoir agent class. After that, HydroCNHS simulates the initial subbasin runoffs independently using a rainfall-runoff module, for which we provide two options: (1) the general water loading function (GWLF; Haith and Shoenaker 1987) with nine parameters; and (2) the ABCD model (Thomas 1981) with five parameters. Next, we use the Lohmann routing model (Lohmann et al. 1998) to trace the runoff from subbasins through the river channel (i.e., inter-subbasin routing) and the unit hydrograph parameterization described in Wi et al. (2015) to account for the within-subbasin routing process. A detailed description of the GWLF, ABCD, and Lohmann routing models is provided as supplemental information in Appendix S1. The runoff from each subbasin is sent to the "Coupling Simulation" step (Fig. 1), and its contribution to the basin outlet is determined by the three factors: (1) the simulation period, (2) the routing scheme for routing outlets, and (3) agents linked to outlets. Forming a routing scheme with four APIs in HydroCNHS will be further explained in the following sections.

The GA "Calibration" module, powered by the Distributed Evolutionary Algorithms in Python (DEAP) Python package (Fortin et al. 2012; De Rainville et al. 2012), facilitates calibrating the entire CNHS model in a parallel computing mode. We refer our readers to the HydroCNHS user manual for more details and coding examples (https://hydrocnhs.readthedocs.io).

### Routing Scheme

The routing scheme assigns an order to each routing outlet. The routing modules are executed in order from upstream to downstream basins to ensure that the effects of upstream agent properties propagate further downstream explicitly. Note that the topographical network of outlets is predefined in the model configuration file by users. The routing outlets are where the streamflow information is required for calibration or agents' decisions (e.g., reservoir release rules). The backtracking process automatically generates the routing scheme starting from the basin outlet (e.g., N1 in Fig. 2). Moreover, the HydroCNHS supports multibasin simulation (e.g., N1 and n1 in Fig. 2) for transboundary analysis. The routing scheme in Fig. 2 is expressed as [N4, N5, <u>R2</u>, R1, N3, <u>N2</u>, N1, n1]. This sequence will be adjusted accordingly if users add specific "node groups" in the model configuration file. For instance, if the release rule of the reservoir R2 is influenced by the streamflow at N2, we need to acquire streamflow at the location before making a decision on the release from R2. This sequence can be refined by assigning a "node group" to N2 and R2 in the model

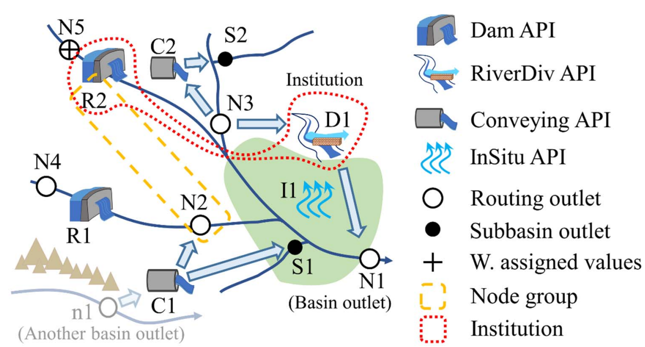


**Fig. 2.** Generic example of HydroCNHS coupling APIs and water system description. Note that agents R1, R2, D1, C1, C2, and I1 are programmed in ABM modules (*·py*) and integrated into HydroCNHS through APIs.



J. Water Resour. Plann. Manage., 2022, 148(12): 06022005

configuration file. After that, the output routing scheme will be automatically updated in the HydroCNHS as [N4, R1, N5, N2, R2, N3, N1, n1].

### Coupling APIs

APIs, herein, are the communication interface between HydroCNHS and user-defined ABM modules. The four APIs in the HydroCNHS (Fig. 2) are (1) Dam API, (2) RiverDiv API, (3) Conveying API, and (4) InSitu API. **Dam API** is designed for integrating in-stream agents like reservoirs (e.g., R1 and R2 in Fig. 2) that could significantly alter the streamflow regime. Agents with Dam API will be considered as pseudo–routing outlets (no routing is needed) involved in the routing scheme. Namely, streamflow is directly defined by agents' water release decisions. **RiverDiv API** is created for agents that divert water from rivers and may have return flows to other outlets, e.g., diversion agent D1 diverts water from N3 and return water to N1 in Fig. 2. This API ensures the diverted outlet is routed before agents' diversions. At the outlet receiving return flow, the subbasin runoff and returned flow are combined and enter the within-subbasin routing process, since return flows often have no explicit return locations. **Conveying API** is designed to transfer water to another outlet from a routing outlet where the routing process has already been executed. The transferred water has no within-subbasin routing (no within-subbasin delay like runoff). Therefore, they are routed separately from the subbasin's runoffs. If an agent wants to convey water from the downstream outlet to the upstream outlet (e.g., pump stations), the water will be delivered with delays (e.g., C2 diverts water from N3 first and delivers it to S2 at a later time step). **InSitu API** is developed for agents that directly affect runoffs via "within subbasin activities" (e.g., I1 in Fig. 2). For example, those runoff changes may come from land-use changes due to urbanization or exploiting groundwater through wells. Such adjustments will be made before any routing process at each time step.

We mathematically formalize the coupled model simulation with these four APIs at each time step. Eq. (1) shows all runoff components within a subbasin before routing

$$F'_s = F_s + \sum_{g\in \mathrm{Agt}_I(s)} Eu_g^+ + \sum_{g\in \mathrm{Agt}_I(s)} Eu_g^- + \sum_{g\in \mathrm{Agt}_R(s)} Re_g^+, \text{ for all } s \in \{\text{subbasins}\} \tag{1}$$

where $F_s$ and $F'_s$ = initial and updated runoff in subbasin $s$, respectively; $Eu$ = runoff changes with symbols of plus (gain) and minus (loss) for InSitu agents $\mathrm{Agt}_I(s)$ activated at outlet $s$; $\mathrm{Agt}_R(s)$ = RiverDiv agents activated at outlet $s$; and $Re$ = return flow.

The node-to-node routing is simulated using Eqs. (2) and (3). Eq. (2) represents the streamflow replacement by in-stream agents like reservoirs

$$Q_g = f_{\text{release}}(g), \text{ for all } g \in \{\text{agents using Dam API}\} \tag{2}$$

where $Q_g$ = streamflow expressed as a function $f_{\text{release}}(\cdot)$, taking the Dam agent $g$ as inputs. Eq. (3) computes all other routing processes and streamflow changes resulting from conveying flow and diversions

$$Q_r = \sum_{s\in F_u, u\in U(r)} f_{\text{rout}}(F'_s) + f^I_{\text{rout}}(F'_r) + \sum_{a\in A(r)} f^R_{\text{rout}}(F'_a) + \sum_{g\in \mathrm{Agt}_C(r)} f^R_{\text{rout}}(C_g^+) + \sum_{g\in \mathrm{Agt}_C(r)} C_g^- + \sum_{g\in \mathrm{Agt}_R(r)} D_g^-, \text{ for all } r \in \{\text{routing outlets}\} \setminus \{\text{agents using Dam API}\} \tag{3}$$

where $Q_r$ = routed streamflow at routing outlet $r$; $f_{\text{rout}}(\cdot)$ = the Lohmann routing function; $f^I_{\text{rout}}(\cdot)$ and $f^R_{\text{rout}}(\cdot)$ = routing functions considering only within-subbasin routing and inter-subbasin (i.e., river) routing, respectively; $U(r)$ = upstream outlets contributing to the streamflow at $r$; $A(r)$ = routing outlets with assigned streamflow time series for which within-subbasin routing is not required; $\mathrm{Agt}_C(r)$ and $\mathrm{Agt}_R(r)$ = Conveying agents and RiverDiv agents activated at routing outlet $r$, respectively; $C$ = conveying water; and $D$ = diversion. The plus and minus signs indicate flow changes due to adjustments. Each agent has a priority input for the simulation order in case conflicts occur (e.g., diversions at an outlet by multiple agents). Also, HydroCNHS supports the institution feature in which multiple agents share a decision-making instance/object allowing them to make decisions together. For example, R1 and D1 in Fig. 2 coordinate on the release and diversion decisions.

## Case Study: Tualatin River Basin

We selected the TRB as a study area (Fig. 3) to demonstrate the four APIs in HydroCNHS. The TRB, located in northwest Oregon, United States, with a drainage area of 1,844.07 km$^2$, is covered by densely populated areas (20%), agricultural area (30%), and forests (50%) (Tualatin River Watershed Council 2021). Its agriculture heavily relies on the irrigation scheme accounting for high seasonal rainfall variability, because rainfall in the area concentrates during the winter season (November–February). The Spring Hill Pumping Plant is the largest diversion facility in the TRB for supporting the Tualatin Valley Irrigation District (TVID; **DivAgt**), where the Hagg reservoir (**ResAgt**) is the primary water source. During the summer period, water is transferred from the Barney reservoir (outside of the TRB) through a transbasin aqueduct (**PipeAgt**) to augment the low flow for ecosystem conservation.

We modeled seven TRB subbasins (in HydroCNHS) and three agents (in an externally programmed TRB_ABM module). The seven outlets of subbasins are denoted TRTR, $\mathrm{Hagg_{In}}$, DLLO, TRGC, DAIRY, RCTV, and WSLO (Fig. 3). The three agents are PipeAgt, ResAgt, and DivAgt, integrated through Conveying, Dam, and RiverDiv APIs, respectively (Fig. 3). PipeAgt (i.e., a water manager) assigns conveying water to TRTR with observed median monthly values (Bonn 2020). ResAgt (i.e., a reservoir operator) determines reservoir releases with generic operational rules, where target storages and target releases are adopted for flood control (October–May) and storage control (June–September) periods. DivAgt (i.e., a group of farmers) diverts water from TRGC with monthly diversion-request decisions at the beginning of each month and has return flow to WSLO. The diversion-request

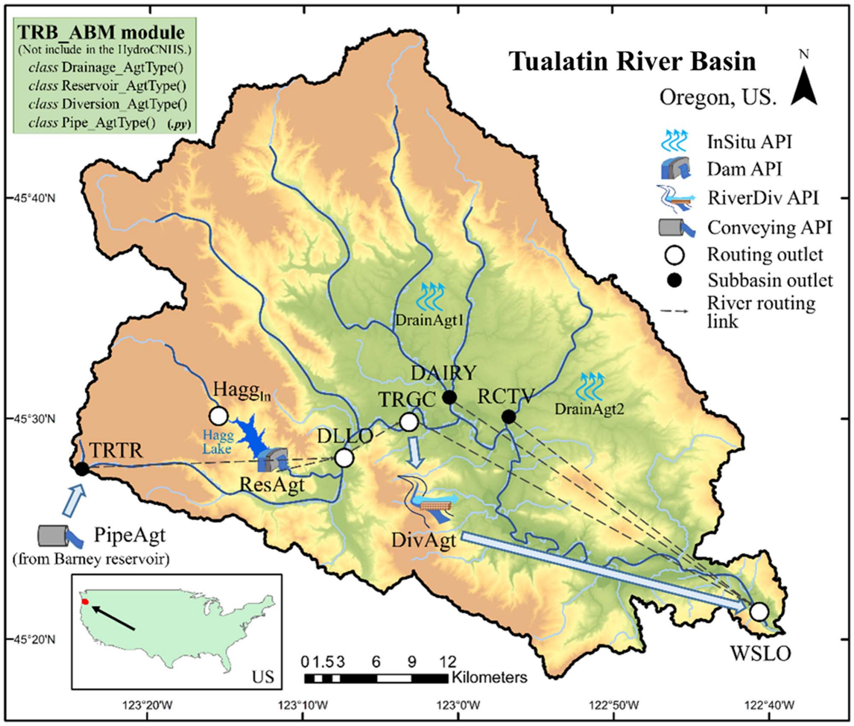


**Fig. 3.** The Tualatin River Basin system. PipeAgt, ResAgt, and DivAgt are transbasin aqueduct, Hagg reservoir, and TVID agents, respectively. DrainAgt1 and DrainAgt2 are two drainage system agents for the runoff-changing scenario.

**Table 1.** KGE comparison for the calibration and validation results of two models

| Model | ResAgt | DLLO | DivAgt | WSLO |
|---|---|---|---|---|
| Monthly observed data | Reservoir releases | Streamflow | Diversion | Streamflow |
| $M_{gwlf}$ | (0.783, 0.811) | (0.916, 0.865) | (0.917, 0.898) | (0.958, 0.894) |
| $M_{abcd}$ | (0.776, 0.893) | (0.905, 0.889) | (0.905, 0.885) | (0.777, 0.836) |

Note: calibration 1981–2005, validation 2006–2013, on a monthly scale.

decisions from June to September are governed by linear functions, where the observed monthly precipitation is the predictor. Minor diversions in other months are filled with historical mean values. Details of this TRB_ABM module and agents' decision rules are provided in Appendix S2.

We tested the simulation from 1981 to 2013, for which the climate data were obtained from Livneh et al. (2015). We aggregated the 1/16-degree climate grids for each subbasin and agent. The HydroCNHS GA module conducts the calibration (1981–2005) with Kling-Gupta efficiency (KGE; Gupta et al. 2009) as a target performance metric. We compared two models, $M_{gwlf}$ and $M_{abcd}$, in which the same ABM model was coupled with two rainfall-runoff modules, GWLF and ABCD, respectively. Detailed calibration settings, including calibration objective and data sources, are provided in Appendix S3. In addition, we ran a scenario with fixed diversion behavior of DivAgt using a monthly mean (i.e., a conventional method to handle human decisions exogenously) to compare with endogenous adaptive behavioral rules. Such differences in human behavior assumptions may lead to distinct modeling outcomes and impact the exploratory analysis of changing environments. To demonstrate the usage of InSitu API, we ran runoff-changing scenarios with the calibrated $M_{gwlf}$ model to test how the changes in upstream runoff affect the streamflow at the basin outlet. One possible cause of runoff changes is urbanization. Therefore, we modeled runoff changes by adding two agents, **DrainAgt1** at DAIRY and **DrainAgt2** at RCTV, and assumed a linear growth of the urbanized area in DAIRY and RCTV subbasins from 5% to 50%, where such urbanization is assumed to increase unit runoff by 75% according to a local study (Gwenzi and Nyamadzawo 2014).

## Results

We compare KGEs between two calibrated models ($M_{gwlf}$ and $M_{abcd}$) in Table 1 and Fig. S1. The two values in parentheses are KGEs for calibration (1981–2005) and validation (2006–2013) periods, respectively. Both models can capture streamflow dynamics and agent behaviors (i.e., reservoir releases and water diversions) on a monthly scale. $M_{gwlf}$ has better performance in general because the GWLF model uses nine parameters for each subbasin compared with the five-parameter ABCD model. One advantage of endogenous behavioral rules is that they can capture the dynamic interactions between natural and human systems and more realistically present the variances of two systems under the changing environment. For example, Fig. 4(a) shows the difference in the annual outputs' variance between $M_{gwlf}$ with the calibrated endogenous diversion behavioral rules ($M_{gwlf,endog}$) and $M_{gwlf}$ with fixed diversion ($M_{gwlf,fixed}$). $M_{gwlf,endog}$ has a larger variance in DivAgt's diversion, reflecting that DivAgt (and farmers) can adjust water diversion according to the weather forecast. Such adaptive diversion

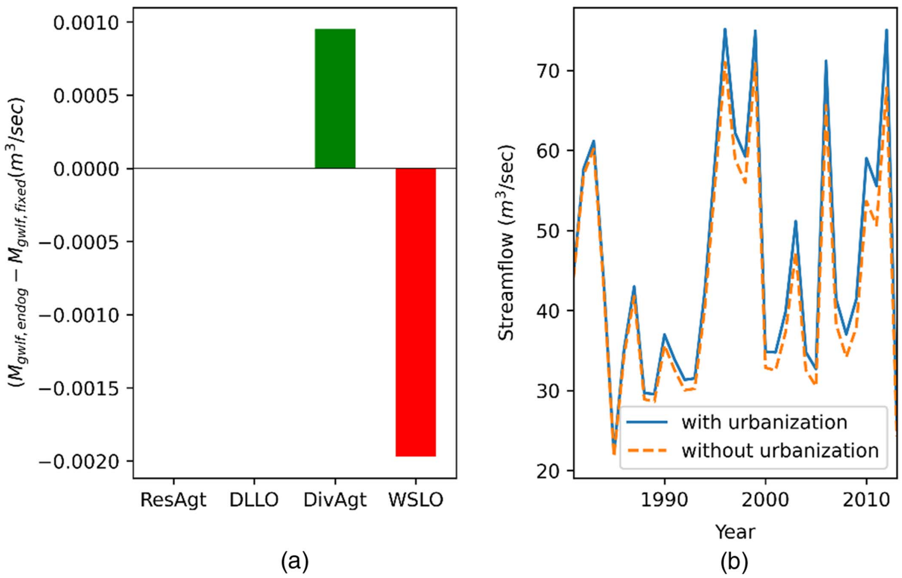


**Fig. 4.** Two scenarios of the $M_{gwlf}$ model: (a) standard deviation difference between $M_{gwlf,endog}$ and $M_{gwlf,fixed}$ in annual mean values; and (b) the annual WSLO streamflow with (solid line) and without (dashed line) urbanization using calibrated model.

behaviors counteract the streamflow at the downstream, leading to lower streamflow variance at WSLO [Fig. 4(a)] compared with $M_{gwlf,fixed}$. $M_{gwlf,endog}$ may help exploratory analysis in which the environment gradually changes and static behavioral rules (e.g., $M_{gwlf,fixed}$) are no longer appropriate. To demonstrate the last API (i.e., InSitu API), Fig. 4(b) shows the gradual annual streamflow increment (gaps between two lines) at WSLO resulting from the runoff changes at DAIRY and RCTV subbasins.

## Conclusions

This technical note presents a semidistributed hydrological model for coupled natural–human systems, HydroCNHS, an open-source Python package. We demonstrate the functionalities of HydroCNHS through a case study in the Tualatin River Basin, Northwest United States, where we coupled a trans-basin aqueduct, a reservoir, an irrigation diversion, and two drainage system agents accounting for runoff changes with four coupling APIs linked to two different rainfall-runoff models, GWLF and ABCD. The KGE comparison results indicate that coupled models could capture monthly streamflow, irrigation diversion, and reservoir release patterns. We also show that the model with an endogenous diversion behavioral rule better reflects the interaction between natural and human systems and may facilitate exploratory analysis. Also, the results of the runoff-changing scenario show the capability of HydroCNHS in modeling the effects of gradual environmental changes on streamflow. With coding language integrity, flexibility in designing agents, and parallel computing ability, HydroCNHS can potentially benefit future studies in CNHS such as uncertainty analysis or coupling of agent designs that are more diverse (e.g., hydropower plants and cooling plants) and complex (e.g., interactions among agents and hydrological environment).

## Data Availability Statement

All data, models, and code generated or used during the study appear in the published article. The HydroCNHS Python package was developed under Python 3.8. The code, user manual, and TRB example (including input data) can be downloaded at https://github.com/philip928lin/HydroCNHS. The weather data originated from Livneh et al. (2015). The streamflow and Hagg reservoir were obtained from the US Bureau of Reclamation Hydromet platform (https://www.usbr.gov/pn/hydromet/tuatea.html) and Bonn (2020). Detailed station information is provided in Table S1.

## Acknowledgments

The work described in this paper was supported by the US National Science Foundation (NSF): CBET #1941727. We thank the editor, the associate editor, and two anonymous reviewers for their comments and suggestions to improve the quality of the manuscript.

## Supplemental Materials

Appendixes S1–S3, including Eqs. (S1)–(S37), Tables S1–S3, and Fig. S1, are available online in the ASCE Library (www.ascelibrary.org).

## References

Arnold, J. G. et al. 2012. “SWAT: Model use, calibration, and validation.” *Trans. ASABE*. 55 (4): 1491–1508.

Bonn, B. 2020. *Tualatin River flow management technical committee 2020 annual report*. Hillsboro, OR: Clean Water Services.

De Rainville, F. M., F. A. Fortin, M. A. Gardner, M. Parizeau, and C. Gagné. 2012. “Deap: A python framework for evolutionary algorithms.” In *Proc., 14th Annual Conf. Companion on Genetic and Evolutionary Computation*, 85–92. New York: Association for Computing Machinery.

Faust, K. M., D. M. Abraham, and D. DeLaurentis. 2017. “Coupled human and water infrastructure systems sector interdependencies: Framework evaluating the impact of cities experiencing urban decline.” *J. Water Resour. Plann. Manage.* 143 (8): 04017043. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000794.

Fortin, F. A., F. M. De Rainville, M. A. G. Gardner, M. Parizeau, and C. Gagné. 2012. “DEAP: Evolutionary algorithms made easy.” *J. Mach. Learn. Res.* 13 (1): 2171–2175.

Gupta, H. V., H. Kling, K. K. Yilmaz, and G. F. Martinez. 2009. "Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modeling." *J. Hydrol.* 377 (1–2): 80–91. https://doi.org/10.1016/j.jhydrol.2009.08.003.

Gwenzi, W., and G. Nyamadzawo. 2014. "Hydrological impacts of urbanization and urban roof water harvesting in water-limited catchments: A review." *Environ. Processes* 1 (4): 573–593. https://doi.org/10.1007/s40710-014-0037-3.

Haith, D. A., and L. L. Shoenaker. 1987. "Generalized watershed loading functions for stream flow nutrients 1." *JAWRA J. Am. Water Resour. Assoc.* 23 (3): 471–478. https://doi.org/10.1111/j.1752-1688.1987.tb00825.x.

Hu, Y., and S. Beattie. 2019. "Role of heterogeneous behavioral factors in an agent-based model of crop choice and groundwater irrigation." *J. Water Resour. Plann. Manage.* 145 (2): 04018100. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001033.

Knox, S., P. Meier, J. Yoon, and J. J. Harou. 2018. "A python framework for multi-agent simulation of networked resource systems." *Environ. Modell. Software* 103 (Jun): 16–28. https://doi.org/10.1016/j.envsoft.2018.01.019.

Liang, X., E. F. Wood, and D. P. Lettenmaier. 1996. "Surface soil moisture parameterization of the VIC-2L model: Evaluation and modification." *Global Planet. Change* 13 (1–4): 195–206. https://doi.org/10.1016/0921-8181(95)00046-1.

Lin, C.-Y., and Y.-C. E. Yang. 2022. "The effects of model complexity on model output uncertainty in co-evolved coupled natural-human systems." *Earth's Future* 10 (6): e2021EF002403. https://doi.org/10.1029/2021EF002403.

Lin, C.-Y., Y.-C. E. Yang, K. Malek, and J. C. Adam. 2022. "An investigation of coupled natural human systems using a two-way coupled agent-based modeling framework." *Environ. Modell. Software* 155 (Jun): 105451. https://doi.org/10.1016/j.envsoft.2022.105451.

Livneh, B., T. J. Bohn, D. W. Pierce, F. Muñoz-Arriola, B. Nijssen, R. Vose, D. R. Cayan, and V. Brekke. 2015. "A spatially comprehensive, meteorological data set for Mexico, the U.S., and southern Canada (NCEI Accession 0129374). [The year 1981 to 2013]." Accessed November 7, 2021. https://cawaterlibrary.net/document/a-spatially-comprehensive-hydrometeorological-data-set-for-mexico-the-u-s-and-southern-canada-1950-2013/.

Lohmann, D., E. Raschke, B. Nijssen, and D. P. Lettenmaier. 1998. "Regional scale hydrology: I. Formulation of the VIC-2L model coupled to a routing model." *Hydrol. Sci. J.* 43 (1): 131–141. https://doi.org/10.1080/02626669809492107.

NASEM (National Academies of Sciences, Engineering, and Medicine). 2021. "Developing a toolkit for fostering open science practices." In *Proc., Workshop*. Washington DC: National Academies Press.

Sivapalan, M., and G. Blöschl. 2015. "Time scale interactions and the coevolution of humans and water." *Water Resour. Res.* 51 (9): 6988–7022. https://doi.org/10.1002/2015WR017896.

Thomas, H. A. 1981. *Improved methods for national water assessment*. Report, Contract: WR15249270. Washington, DC: US Water Resource Council.

Tualatin River Watershed Council. 2021. "Tualatin basin information." Accessed November 19, 2021. http://trwc.org/tualatin-basin-information/.

Wada, Y., et al. 2017. "Human–water interface in hydrological modelling: Current status and future directions." *Hydrol. Earth Syst. Sci.* 21 (8): 4169–4193. https://doi.org/10.5194/hess-21-4169-2017.

Whitley, D. 1994. "A genetic algorithm tutorial." *Stat. Comput.* 4 (2): 65–85. https://doi.org/10.1007/BF00175354.

Wi, S., Y. C. E. Yang, S. Steinschneider, A. Khalil, and C. M. Brown. 2015. "Calibration approaches for distributed hydrologic models in poorly gaged basins: Implication for streamflow projections under climate change." *Hydrol. Earth Syst. Sci.* 19 (2): 857–876. https://doi.org/10.5194/hess-19-857-2015.

Yang, Y. E., K. Son, F. Hung, and V. Tidwell. 2020. "Impact of climate change on adaptive management decisions in the face of water scarcity." *J. Hydrol.* 588 (Sep): 125015. https://doi.org/10.1016/j.jhydrol.2020.125015.