Manuscript title:

Enabling Collaborative Numerical Modeling in Earth Sciences using Knowledge Infrastructure

Authors:

C. J. Bandaragoda[1*], A. Castronova[2], E. Istanbulluoglu[1], R. Strauch[3], S. S. Nudurupati[1], J. Phuong[4], J. M. Adams[5,6], N. M. Gasparini[7], K. Barnhart[8], E. W. H. Hutton[6], D. E. J. Hobley[9], N. J. Lyons[7], G. E. Tucker[6,8], D.G. Tarboton[10], R. Idaszak[11], S. Wang[12]

Author affiliation:

[1] Department of Civil and Environmental Engineering, University of Washington, Seattle, USA.

[2] Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI), USA. [3] Seattle City Light, Seattle, USA. [4] Department of Biomedical Informatics and Medical Education, University of Washington, Seattle, USA. [5] Institute of Arctic and Alpine Research, University of Colorado, Boulder, Colorado USA. [6]Community Surface Dynamics Modeling System (CSDMS), University of Colorado, Boulder, USA. [7]Department of Earth and Environmental Sciences, Tulane University, New Orleans, LA, USA. [8]Department of Geological Sciences, University of Colorado, Boulder, USA. [9]Cardiff University, Cardiff, UK. [10]Department of Civil & Environmental Engineering, Utah State University, Logan, USA. [11] Renaissance Computing, University of North Carolina, Chapel Hill. [12]Department of Geography and Geographic Information Science, University of Illinois at Urbana-Champaign, Urbana, USA.

*Correspondence should be addressed to Dr. Christina Bandaragoda. (cband@uw.edu), Civil & Environmental Engineering, University of Washington, 201 More Hall, Box 352700, Seattle, WA 98195-2700.

Highlights:

- Knowledge Infrastructure can address common challenges to using online systems for collaborative numerical modeling.
- Advanced cyberinfrastructure coupled with research community support supports all users in efficiently advancing Earth system sciences.
- Three computational narratives using Landlab on HydroShare demonstrate how to replicate and reuse Earth surface models for education.

Keywords: cyberinfrastructure, knowledge infrastructure, reproducible modeling, Landlab, HydroShare, education

Software and/or data availability:

- Models described in this manuscript are available on HydroShare. Citation: Bandaragoda, C., A. M. Castronova, J. Phuong, E. Istanbulluoglu, S. S. Nudurupati, R. Strauch, N. Gasparini, E. Hutton, G. Tucker, D. Hobley, K. Barnhart, J. Adams (2018). Enabling Collaborative Numerical Modeling in Earth Sciences using Knowledge Infrastructure: Landlab Notebooks, HydroShare, http://www.hydroshare.org/resource/70b977e22af544f8a7e5a803935c329c
- Landlab can be installed from conda-forge. with Windows 7+, Mac OS 10.6+, or Ubuntu LinuxOGH v.1.5.4 is released on GitHub (https://github.com/landlab/landlab/releases), and is freely available under an MIT license. This GitHub repository is maintained by the Landlab development team.
- The Landlab Python library is also vailable within a JupyterHub-Unix docker environment hosted on the CUAHSI HydroShare server.
- Tutorial/Use-case notebooks for developers can be found at the GitHub repository (https://github.com/ChristinaB/tutorials) and the HydroShare resource (http://www.hydroshare.org/resource/70b977e22af544f8a7e5a803935c329c)

Abstract

Knowledge Infrastructure is an intellectual framework for creating, sharing, and distributing knowledge. In this paper, we use Knowledge Infrastructure to address common barriers to entry to numerical modeling in Earth sciences: computational modeling education, replicating published model results, and reusing published models to extend research. We outline six critical functional requirements: 1) workflows designed for new users; 2) a community-supported collaborative web platform; 3) distributed data storage; 4) a software environment; 5) a personalized cloud-based high-performance computing platform; and 6) a standardized open source modeling framework. Our methods meet these functional requirements by providing three interactive computational narratives for hands-on, problem-based research demonstrating how to use Landlab on HydroShare. Landlab is an open-source toolkit for building, coupling, and exploring two-dimensional numerical models. HydroShare is an online collaborative environment for the sharing of data and models. We describe the methods we are using to accelerate knowledge development by providing a suite of modular and interoperable process components that allows students, domain experts, collaborators, researchers, and sponsors to learn by exploring shared data and modeling resources. The system is designed to support uses on the continuum from fully-developed modelling applications to prototyping research software tools.

1 Introduction

Modeling in Earth sciences began with the use of hand-written mathematical formulas that were developed from observational evidence, conjecture, or hypothesis, and shared through conversation and correspondence. As richness and complexity of our available Earth observations have grown in parallel with technological advances in computational resources (supercomputing, high-performance computing, and cloud computing), our models now focus on couplings among atmospheric, hydrologic, ecologic, geomorphic and human-impacted processes (e.g., Tucker and Hancock, 2010; Yetemen et al., 2015a, b; Han et al., 2014, Anders et al., 2008; Pande and Sivapalan, 2016). Advances in internet-based cyberinfrastructure research tools and technology, also broadly considered as Knowledge Infrastructure (KI), have expanded our capacity for structured collaborations in research (Edwards et al., 2013). However, these advances often come at the expense of raising the technological bar for entry into numerical modeling. Here, with examples from Earth science, we discuss these advances as enablers that include three key features: 1) a community platform that allows dynamic interactions among developers, researchers, and new users; 2) clear documentation of theoretical and mathematical details that are often lost for new users of complex model programs; and 3) model reproducibility. For example, sharing the code and data within a community portal with computational capacity allows new users to easily find and test training materials, developers to easily distribute open workshop materials, and communities to build new research networks. Further, as technology is integrated in the research with greater sophistication, it is increasingly a

challenge to keep the fundamental equations that define the driving assumptions in the model structure accessible to software users. Using methods such as inclusion of equations and references in online documents, can avoid the 'black box' syndrome and improve ease of learning, transparency, and usability of the modeling code. This provides the Domain of Applicability (Netzeva et al., 2005), or 'building ignorance into the system (Edwards et al., 2018) to be clear on the purpose and limits of the model. Experimental design is addressed by illustrating three different areas where model reproducibility can have an impact on advancing science: classroom and peer-to-peer education, replicating published results, and reusing models to build new research products.

Knowledge Infrastructure (KI) is an emerging intellectual framework to understand and improve how people create, share, interpret observations and modeled results, and distribute knowledge, which has dramatically changed and is continually transformed by internet technologies. KIs are most simply defined as "robust networks of people, artifacts, and institutions that generate, share, and maintain specific knowledge about the human and natural worlds" (Borgman and Traweek, 2012). In Earth and hydrologic sciences, interpreting observational and model simulated data is a fundamental task, but systematic acquisition for interpretations and machine readability is not common practice among environmental research infrastructures (Stocker et al., 2018). KI advances us beyond cyberinfrastructure, which is limited to distributed computer, information, and communication technologies, by including networks of groups and institutions, and the cultural practices of developing and sharing computational narratives (Brooks, 1997; Perez and Granger, 2015). Computational narratives are the algorithmic processes involved in creating and interpreting computed representations (Mani, 2013). In our case, the algorithmic processes are Earth surface models, and the computed representation is the results of the analytical research and how those results are summarized. Recent developments in the use of advanced cyber-infrastructure in Earth science include tools used to support hydroinformatics, such as HydroShare (Tarboton et al., 2014a; Tarboton et al., 2014b; Tarboton et al., 2018) and the CUAHSI JupyterHub service (Castronova, 2017; Perez and Granger, 2015). Development efforts concentrate on the application of information and communication technologies (ICTs) targeted for geospatial analytics (Yin et al., 2017) and hydrologic data types and models (Horsburgh et al., 2016; Morsy et al., 2017, Strauch et al., 2018) by providing resources for open-source practices, including sharing of data and models, and providing cloud computing. With expert knowledge or user experience designed to support non-experts, these platforms can be effectively used for expanding and broadening our capacity to investigate hydrologic and Earth system processes.

In model-based investigations, reproducibility increases confidence in results and improves interpretation about what results do and do not mean; lack of reproducibility limits the expansion and growth of knowledge (Hutton et al., 2016; Nosek et al., 2015). The advancement of knowledge and lowering the barriers to reproducibility can be enabled by KI that supports

collaborative research, education, and curriculum development, and improves standards for technology practices for publication of research and the description of research presented in journal articles. Web-based interactive computing environments, such as Jupyter Notebooks (Perez and Granger, 2015), are designed to execute models and perform data analytics, and have become increasingly prevalent to improve reproducibility of research in the past few years (Shen, 2014), especially with early adopters in the biological sciences (Gross et al., 2014; Ragan-Kelley et al., 2013; Ding and Schloss, 2014). Francis (2018) created a reference of 36 Jupyter Notebooks currently available on the web, with limited examples of experiments in the Earth sciences community. One of the first interactive notebooks that we know of was published by Shen (2014), which provided computational resources for executing code snippets exploring astronomy data. This was available online as an interactive Notebook for three years (2014-2017), and it was replaced with a static view of an example execution of the Notebook in November 2017. Luo et al. (2016) have shown that interacting with web-based models (through a graphical user interface) in classroom environments improves higher-level thinking and attitudes about complex landscape evolution models. We do not know of any collections of Notebooks published with the supporting infrastructure available for authors to maintain accessible interactive Notebooks for their readers in hydrologic sciences and Earth surface modeling communities, or studies on how interacting with model code improves educational or research outcomes.

Recent efforts to provide online computing capacity for Earth science research and education included landscape evolution modeling such as the WILSIM model (Luo et al., 2004; 2016), and watershed hydrology and erosion modeling using WEPP (Laflen et al., 1991; Laflen et al., 1997; Flanagan et al., 2001). While these recent web-based modeling approaches lower the bar for model execution, model options and the level of interaction of the users with models are constrained by the limited to the set of options envisioned by the developers. These tools rely on graphical user interfaces (GUI) with limited user inputs (parameter values, or scenario choices), but they do not provide an interactive software environment for user collaboration and co-creation of knowledge. To develop a persistent, collaborative environment that will have a profound transformational effect on our society (Newman et al., 2003), we need to identify and overcome the barriers that are currently preventing rapid adoption of Knowledge Infrastructure for Earth surface modelers.

This paper is motivated by the following questions: *Can current software infrastructure and research communities (1) facilitate rapid adoption and scientific advancement of complex Earth surface models, (2) lower the bar for entry into modeling, (3) improve collaborations among scientists and science partners, and (4) to develop usable science and sustainable open source software?* In addressing these questions, our aim is to explore Knowledge Infrastructure using advanced data access and computational resources beyond what an individual scientist would normally have available (Bandaragoda et al., 2006). Below we first describe three emerging

open-source modeling practices to lower the bar into modeling (Section 2). In our methods, (Section 3), we review basic elements of KI and substantiate it with specific technical details as implemented in HydroShare, the CI platform we use in this study. In the results (Section 4), we focus on the use of the Landlab Earth surface modeling toolkit (Hobley et al., 2017), deployed on HydroShare, in three use cases that employ emerging open-source modeling practices to lower barriers in modeling with specific workflows designed with interactive notebooks aimed at Earth science education, and reuse and replication of a research model. Our discussion (Section 5) explores the barriers we have identified, followed by Conclusions (Section 6) on our approach to address the motivating questions and current limitations.

2 Methods

2.1 Emerging practices for modeling

The use of cyberinfrastructure to reproduce experiments and share data is expanding. Open source cyber-infrastructure platforms for research publication are designed to facilitate the use of existing models by making input data and model code publicly available online and providing software tools for pre- and post-processing data, running models, sharing data, and formally publishing with a digital object identifier (Freeman, 2005; Atkins, 2003). Using recent examples in water monitoring (Horsburgh et al., 2017; Jones et al., 2017; Mihalevich, B.A. 2017), landslide modeling (Strauch et al., 2018), and data science (Freire et al., 2016), we identified three critical open-source technology practices supported by KI expected to scientific discoveries:

2.1.1 Code development in an open source environment

Evolving software versions, hardware requirements, numerical methods, and code quality limit the ability to replicate and reuse model applications. Developing models from a personal computer (PC) requires installing a suite of specialized software tools and access to computational hardware to visualize, store, and prepare model inputs and outputs. Thus, reproducing a study by others often depends on the ability to reproduce the software environment.

2.1.2 Cyber-training in numerical modeling education

The use of numerical models for science education should not diminish the instruction time for basic science. Costs that sometimes arise from using models in the classroom include time needed for extensive technological instruction, and technical troubleshooting. These costs can be avoided by developing software infrastructure that accesses computational and data intensive models from a web browser, thereby avoiding the need for any software installations, enabling classroom experiences for students that improves understanding of existing theory, and generates curiosity to propose hypotheses and design further modeling, field, and laboratory experiments.

2.1.3 Cyber-interactions in Collaboration

In most research projects, skills for code development, diagnostics, and model execution are limited to a few individuals (graduate students, postdocs, etc). However, most modelers would agree that coding errors can be more effectively identified, more user-friendly codes can be designed, and new research ideas can be developed when other experts have access to models for evaluation, experimentation, and testing. Therefore, the use of Knowledge Infrastructure for research studies where scientists and stakeholders can interact with, execute, and visualize various components of coupled models used in collaborative projects, can provide a research process for rapid development of ideas and research products, leading to more usable science (Lemos et al. 2012).

2.2 Knowledge Infrastructure Design

Our approach includes the following six methodological, software, and hardware components that can address the barriers to computational modeling: 1) User Experience Design is the conceptual and evolving design of the CI that includes all the practices developed to efficiently accomplish collaborative online tasks based on personal, collaborator, and institutional cultural preferences (e.g. workflow practices for using software to run models and perform data analysis). The user experience design guides the development of the framework, and in research software, includes contributions from both developers and new users; 2) a community-supported collaborative web platform that interacts with a high performance computing (HPC) and data storage nodes, allows for computationally intensive computing, and supports open source publishing and privacy (Section 3.2); 3) data storage that may be distributed to different locations (Section 3.3); 4) a software environment that provides a library of software and programming languages, supporting model applications, version control, data analytics, and facilitates the execution of numerical models (Section 3.4); 5) a cloud-based high performance computing (CHPC) platform that hosts the software environment, models, and personal user space (Section 3.5); and 6) a standardized modeling framework (Section 3.6). The adoption, ongoing adaptation, and growth of an infrastructure system is fundamentally dependent on personal research choices, collaborative dependencies, and institutional policies (Section 3.7).

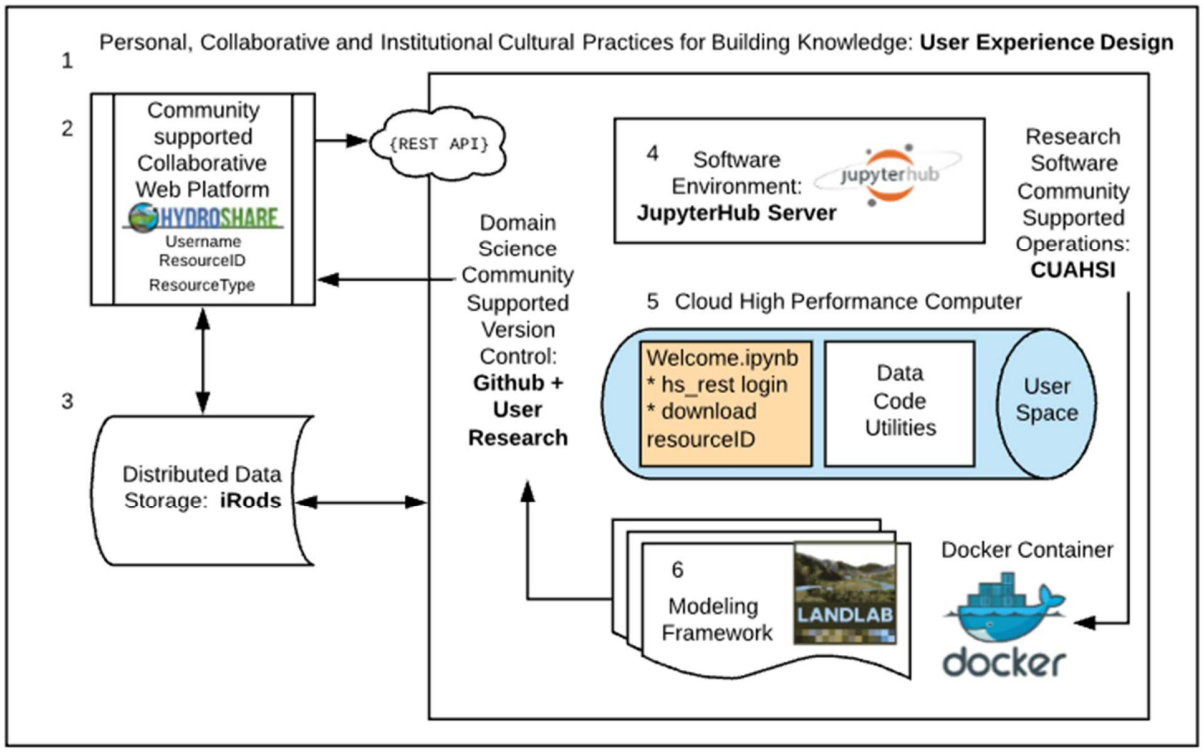


Figure 1. Illustration of six basic elements for a knowledge cyberinfrastructure for interactive community modeling and exploration. Research software communities maintain support of operations between Docker Containers and software environment. Domain science communities maintain support for version control and user communications specific to modeling frameworks.

2.3 Community supported collaborative web platform: HydroShare

HydroShare (www.hydroshare.org) is an online collaborative platform developed to address the growing computer modeling and data storage and sharing needs of the community. It supports the sharing of data and models, developed as HydroShare resources, and facilitates the execution of numerical models deployed on tools, or web apps associated with or linked to HydroShare. HydroShare is operated by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI; www.cuahsi.org) and in our research serves as the Community Supported Collaborative Web Platform (Figure 1, Box 2). A web browser is the interface to HydroShare, which provides access to hydrologic and Earth surface models and data that are saved as resources in HydroShare. The architecture of HydroShare is designed to support: (1) resource storage, (2) resource exploration, and (3) actions on resources. These are implemented using system components that are relatively loosely coupled and interact through APIs. The loose coupling takes advantage of Services Oriented Architecture (SOA) that enhances robustness as components can be upgraded and advanced relatively independently.

2.3.1 Resource Storage

Content that can be shared within HydroShare is diverse, including digital objects that represent multiple hydrologic data types, models and model instances, documents, and other content types commonly used in hydrologic research (Horsburgh et al., 2016). A "resource" is the discrete unit of digital content within HydroShare. Resources are cast as "social objects" that can be published, collaborated around, annotated, discovered, and accessed (Horsburgh et al., 2016). In this resource-centric approach, a resource is the granular unit used for management and access control. HydroShare resources include hydrologic time series, geographic feature (vector data), geographic raster (gridded data), multidimensional space-time data sets (e.g., NetCDF), and composite resources that represent combinations of these data types, as well as collections that group together different resources. Model Programs and Model Instances are additional types of content that can be shared and manipulated within HydroShare. Metadata is maintained that tracks system-level attributes of the resource, including timestamps of creation and modification, ownership, access control rules, etc. Persistent identifiers, access control, versioning, sharing, and discovery are all managed at the resource level in HydroShare. This holds metadata in a standardized and machine-readable way to promote interoperability with other systems. HydroShare's overarching resource data model is an implementation of the Open Archives Initiative Object Reuse and Exchange (OAI-ORE) standard (Lagoze et al., 2008). OAI-ORE is a standard for the description and exchange of web resources. HydroShare uses the Integrated Rule-Oriented Data System (iRODS) (Moore, 2008; iRODS Consortium, 2016; Yi et al., 2018) as its distributed network storage back end. iRODS provides a virtual file system for physical storage distributed across multiple locations and enables data federation across geographically dispersed institutions (Yi et al., 2018).

2.3.2 Resource exploration, discovery and management.

The primary user interface for HydroShare is the website hosted at www.hydroshare.org, developed using the Django web framework (Django Project, 2018) and Mezzanine content management system (Mezzanine Project, 2018). Together, these technologies are used to build a system for archiving data and metadata for each resource; and provides a landing page where metadata can be entered and edited, or content files added or removed. On the landing page users can specify sharing status, e.g. private or public, and manage who has access to edit or view the content. A resource may be permanently published in which case it is precluded from further editing and assigned a citable digital object identifier (DOI). The Django website also provides a "My Resources" page for listing data that belong to or have been shared with each user, a "Discover" page that supports keyword and map based search for content based on their spatial coverage information using the Apaches SOLR search platform (Solr Project, 2018), and a "Collaborate" page for users to create or join groups aligned around specific water themes. Collectively these web pages provide a system where users can discover and manage the content to which they have access, including changing access control settings, and creating

new content. The business logic of resource and content types, and access control is all managed using standard Python Django software packages.

2.3.3 Actions on resources through web apps.

The HydroShare repository, broadly consisting of iRODS middleware for managing data storage and a Django website for content discovery and management, is extended by independent web applications that allow users to perform actions on HydroShare data. Using a services-oriented software pattern, HydroShare has been designed to support this interaction with 3rd-party applications using a Representative State Transfer (REST) application programming interface (API). The industry-standard OAuth protocol is used to manage authentication and interface with HydroShare's access and control model, which is necessary to support interaction with remotely hosted web applications via the API. Flexible web app launching functionality has been established through a HydroShare resource type that defines the URL and parameters for invoking the web application. These web app resources can be created by any HydroShare user to interact with 3rd-party web applications that are designed to act on HydroShare content. HydroShare web applications can be hosted anywhere and have the potential to provide users with a gateway to high performance computing.

2.4 Software Environment: CUAHSI JupyterHub

This paper makes use of a JupyterHub web application developed and maintained by CUAHSI (https://jupyter.cuahsi.org) that leverages Jupyter notebook technology. Jupyter notebooks are an effective way to document research analysis, workflows, and modeling procedures in a reproducible manner (Kluyver et al., 2016). The CUAHSI-JupyterHub service is under active development to support (1) computationally intensive research, (2) data intensive research, (3) education and the dissemination of knowledge, and (4) reproducible science. These goals are made possible through the development of data transfer mechanisms to move data between HydroShare and the JupyterHub environment as seamlessly as possible. Moreover, HydroShare provides a mechanism for users to launch notebook workflows and their associated datasets into a pre-configured, isolated, remote compute environment. Each compute environment is created on-the-fly and contains a persistent data store for performing hydrologic analysis in a manner that is insulated from all other users. This is possible by leveraging operating-system-level virtualization software such as Docker (Merkel, 2014). Each user instance runs the Ubuntu Linux operating system and is pre-configured with scientific Python and R libraries, software for interacting with the HydroShare REST API, and various physical models including as Landlab. A typical workflow is to launch the CUAHSI-JupyterHub web application from a HydroShare resource, programmatically collect any necessary data using the HydroShare REST API, perform modeling and analysis, and finally save results back to HydroShare. After these data, i.e. Jupyter notebook and data files, are saved back to the HydroShare repository, they can be shared with

other users and groups who can further analyze them in a similar way. This back and forth sharing enables collaboration in the development and analysis of Landlab models using the HydroShare repository and linked JupyterHub web app.

2.4.1 Community supported development and operation

CUAHSI supports the development and operation of CUAHSI JupyterHub as part of the HydroShare project (Idaszak et al., 2017) as well as through their cooperative agreement with NSF (see Acknowledgements). Development and operation efforts are divided into two categories, (1) system maintenance and user support, and (2) hydrologic research and modeling. The first category focuses primarily on maintaining existing capabilities, updating libraries, and performing system-level maintenance and upgrades. This includes overseeing the installation and compiling of Python (versions 2.7 and 3.6), R (version 3.4), scientific libraries such as Pandas, Dakota (Adams et al., 2015), SpotPy, NumPy, etc., and modeling applications, e.g. MODFLOW 6, Landlab, TauDEM. The latter category consists of collaborative research to lower the barrier of entry to modern modeling applications such as the Structure for Unifying Multiple Modeling Alternatives (SUMMA) and the National Water Model (NWM) configuration of WRF-Hydro. These efforts are coordinated using an open source codebase in which code contributions undergo a review process and formal release schedule. Users provide feedback and requests via GitHub “bug” and “enhancement” tickets.

2.4.2 Tools and Models

One of the goals of CUAHSI Jupyterhub is to make it simple for users to access the software they need without some of the challenges associated with library dependencies, computer operating system or platform compatibility and installation challenges. As such CUAHSI JupyterHub has installed and supports a range of software and tools commonly used for hydrologic analyses to help users get going quickly in their work; and, make their work more reproducible. It is intended for this set of software and models to grow as the platform is further developed. Currently CUAHSI JupyterHub includes the following

- Landlab, an Earth surface modeling toolkit that this paper focuses on as an example of the approach (Hobley et al., 2017)
- TauDEM, a set of GIS tools for terrain analysis and watershed delineation (Tarboton, 2018; Tesfa et al., 2011)
- MODFLOW Groundwater Model
- The Structure for Unifying Multiple Modeling Alternatives (SUMMA) (Clark et al., 2008; 2011; 2015b; 2015a) model framework that allows for formal evaluation of multiple working hypotheses on model representations of physical processes.
- iRODS iCommands component for accessing large files efficiently from the HydroShare repository using iRODS

- Python tools for working with HydroShare Observation Data Model 2 (ODM2) time series content types (Horsburgh et al., 2016)
- The WaterML R package (Kadlec et al., 2015)

2.4.3 Landlab Community

Landlab has four main release per year (February, May, August, November) which accompany Landlab's quarterly newsletter "The Landlab Lookout". The newsletter alerts users that a new version is available, describes what's new in the release, and gives a summary of Landlab-related news (such as Landlab-themed clinics, publications using Landlab, etc.). Occasionally, intermediate releases will happen in conjunction with annual community meetings that include presentations or workshops that feature Landlab (for instance, American Geophysical Union Annual Meeting (December recurring), Geological Society of America (July recurring), Community Surface Dynamics Modeling System (May recurring)). This ensures that participants of these meetings can use the latest version of Landlab. In addition to announcing new releases via the newsletter, Landlab developers also contact directly other researchers that use Landlab. For HydroShare, this means either submitting issues on the HydroShare JupyterHub Github repository or sending email directly to CUAHSI JupyterHub developers. This ensures that these projects provide their users with the most up-to-date Landlab versions. The role of version control is highlighted in Figure 1, as Domain science community support of research.

2.5 Advanced Cyberinfrastructure and CyberGIS-Jupyter

HydroShare has recently been developed to exploit cyberGIS (that is, geospatial information science and systems based on advanced computing and cyberinfrastructure) and high-performance computing (HPC) (Wang 2010; Wang and Goodchild 2018). CyberGIS-Jupyter allows HydroShare Jupyter notebooks to harness HPC resources such as those provided by the NSF Extreme Science and Engineering Discovery Environment (XSEDE) and Resourcing Open Geospatial Education and Research (ROGER) supercomputer (Wang 2016). Specifically, CyberGIS-Jupyter encompasses the following three major functional components (Yin et al. 2017):

- JupyterHub is used to handle authentication and schedule standalone Jupyter servers. After authentication, dedicated containers are sent to the Docker Swarm.
- Docker Swarm is responsible for spawning and managing all Docker containers across a specific group of virtual machines (the swarm). The containerization provides fine-grain on-demand provisioning of cloud infrastructure as a service when a user launches a notebook.
- Batch HPC is adapted to harness distributed parallel computing resources, high-performance storage systems, and cyberGIS software to greatly expand the capabilities of a typical Jupyter notebook environment.

2.6 Modeling framework: Landlab and its application on HydroShare

A new paradigm in hydrologic and Earth system modeling is emerging where complex systems once coded in Fortran, C++ and cryptic scripts developed for research are being reconfigured in open-source Python, component-based systems. Landlab is one such system based on a Python-language programming library that supports efficient creation and/or coupling of 2D numerical models (Hobley et al., 2017). It is a framework geared towards (but not limited to) Earth-surface dynamics. Landlab is composed of three main divisions of code: grid, components, and supporting utilities. The spatial template for modeling is created by the Landlab ModelGrid class. ModelGrid provides common structured and unstructured (e.g., Voronoi polygons) data structures where data fields can be attached to grid elements, and grid elements can be built as a structured or unstructured grid in a single line of code. Each physical process is coded into individual Landlab class, and added to the Landlab library as a Component, providing an ecosystem of hypothesized behavior of Earth system processes. Supporting utilities and driver scripts were developed to pre-process, post-process, and improve workflow efficiencies for coupling multiple components. Most components operate on, interact with and update grid fields. Components can be coupled via data exchange over the grid. A model driver is a Python script developed to import, instantiate, and run a single or multiple coupled Landlab components. Landlab utilities provide tools for input/output management and visualization. In this paper we use models for coupled ecohydrology and spatial vegetation dynamics, flow routing (Adams et al., 2017), and landslide probability (Strauch et al., 2018) along with a recently developed climate data handling utility. As with the open-source nature of Landlab, a growing community of developers contribute numerical functions, process-based components, and utilities.

A Landlab model developer who is interested to share a Landlab model application can develop their model drivers using a Jupyter notebook. This Jupyter notebook is then deployed on HydroShare, by "publishing" as a resource. Through this process the user obtains a DOI for their resource and the resource becomes available for others to use. Jupyter notebook interacts with the CUAHSI JupyterHub server and executes the model. In preparation of a Landlab JN there are few must-complete steps. These are listed in the pseudo code below, using an example that couples a storm driver, soil moisture, and vegetation dynamics components.

2.7 User experience design for multiple learning pathways

In this section, we describe how the Knowledge Infrastructure can be viewed from the lens of a research workflow presented in Appendix A: using Landlab on HydroShare. Upon publication of a resource and its deployment to users, simply by sharing the location of the resource on HydroShare. Users may be composed of collaborators in a research project, stakeholders of watershed resources, and students, the users begin learning and exploring the code. In the figure

the user interactions with the model depicted in two levels. The user explores Landlab on HydroShare using the deployed model driver by changing parameter values of the process components and perhaps explores other components by adding them to the driver (lower curved-arrow). In the process of exploring the model the user may develop new ideas to develop new process representations presented as new components in Landlab or develop new ways of data visualization. These new developments on Landlab components will require the users to contribute their work to the Landlab repository and Landlab version updates and its further deployment on to HydroShare. These model developments can continue offline by installing and using Landlab on a personal computer, or other JupyterHub servers.

2.8 Data and Models for Three Computational Narratives

To illustrate our methods for lowering the barriers to computational modeling, we have developed three computational narratives for user experience (UX) (Table 1). A computational narrative can be considered a story that can be told about the data by executing scripts that generate data analysis and visualization in the provided workflow. Recognizing that every experience is made of many parts that contribute to the adoption and evolution of tool development, and the narrative (see Section 5, Inductive and/or Deductive) can provide a framework for a user to generate their own story by exploring the science topic with interactive tools. A UX can be described as a computer-human interaction. The importance of UX design is becoming more widely recognized in science and technology development to achieve the desired outcome of the UX such as improving the knowledge-base and cognitive capabilities of users (Baldwin, 2013; Glassdoor, 2017). Given that an experience may be generated by any interaction, we designed three example computational narratives to generate individual experiences, share understanding on existing theory, and to open doors for future developments (Forlizzi and Ford, 2000). In the first case, we give an example of how to use this infrastructure (Figure 1) to develop training and educational materials for classroom curriculum. This example focuses on the use of flexible components in a modeling framework to demonstrate two approaches for flow routing, from simple to more complex solution of the same shallow water equation, with inductive narrative workflows designed to orient new users to a focused set of theoretical concepts that can be explored with minimum background in the computational infrastructure or coding.

In the second case, we illustrate how a researcher may execute a model to replicate reported findings in a published study on annual landslide probability. We particularly focus on how the use of a controlled software environment provides easy access to new users of the tool and facilitates the exploration of other questions associated with the processes investigated using the same tool and data. This is an example of a deductive workflow, where new hypotheses are tested using a published set of tools and data. In the third computational narrative, we present a

more sophisticated example for executing a published ecohydrology model, enhanced and applied in a new location. It uses a component bundling idea for efficient scenario building to explore eco-hydrologic response to a climatic gradient mediated by elevation. This example illustrates a research cycle that includes both inductive and deductive workflows to generate new understanding. Computational narratives demonstrate how to use Knowledge Infrastructure to educate, replicate, and reuse Earth surface models where the user interacts with the infrastructure to develop their own story.

Table 1. Three study problems were designed with a focus to 1) explore, 2) replication, and 3) reuse research. Computational narratives demonstrate how to use Knowledge Infrastructure for computation and visualization of Earth surface models where the user interacts with the infrastructure to develop their own story.

Purpose	Tasks	Highlight	Science Topic	Workflow	Notebook Title
1: Educate	Training and Curriculum Development	Flexible Components	Hydrology	Inductive	explore_flood _routing.ipynb
2: Replicate	Execute a published model to replicate reported findings	Controlled software environment	Landslides and Fire	Deductive	replicate_ landslide_ model.ipynb
3: Reuse	Execute and enhance a published model in a new location	Mulitple components wrapped for efficiency	Ecohydrology	Inductive + Deductive	reuse_ ecohydrology_ gridhydromet.ipynb

Notebooks are designed with up to 10 sections. For example, for the example (see 4.1), Section 1 introduces the theory and the conceptual design of the models. For example, in the first notebook we begin with the theory of the 1-D Saint Venant equation for transient shallow water flow, which is at the core of many hydrodynamic models. Data Science and Cyberinfrastructure methods are provided in Section 2.0, followed by Landlab Methods (Section 3.0). Sections 1 to 3 are designed to function as an interactive textbook or reference. In the section labeled 'Make Model Decisions' (Section 4.0), our aim is to clearly distinguish the component-based options for designing a model experiment. For example, in the first notebook we provide options for designing a storm hydrograph based on the choice of basin, storm intensity, and routing method. Model Computations (Section 5.0) and Results (Section 6.0) provide code to execute the model, visualize results, and export data. Discussion (7.0), Conclusions (8.0), and Saving results to HydroShare (Section 9.0) are designed to support graduate level coursework in Hydrologic Processes and Modeling. Finally, users are provided shell script prompts that can be executed in the Jupyter Notebook to remove data from the JupyterHub server after completing their work (Section 10).

3 Results

In our results we describe three computational narratives we designed to lower barriers to computational modeling using the CI described in Section 2. We relied on Juptyer notebooks for sharing the following computational narratives and designed the sequence of commentary text and code blocks to be generally useful to Earth surface modeling research communities. Hanney and Savin-Badem (2013) suggest that combining project and problem-based learning may be the best practice for generating engagement, critical thinking, and creativity, with the use of problem-based learning as an important tool for providing authentic experiences, highly valued by all learners (Kokotsaki et al., 2016).

3.1 Notebook 1: Exploring runoff hydrographs with Landlab

3.1.1. Notebook 1 Overview

This notebook provides resources to compare two different flow routing schemes, kinematic wave and overland flow (2D de'Almeida (2012) solution of the Saint Venant equation), as explained on the notebook in detail, in two different landscapes for a given rate of rainfall excess (rainfall in excess of infiltration). The notebook can be used to investigate process-based questions on the generation of overland flow hydrographs across the landscape in relation to the role of runoff rate, watershed topography, network structure, and surface roughness, and it allows to compare and contrast the properties of streamflow hydrographs generated by the two different flow routing algorithms. To provide a contrast between different landscape shapes, this notebook uses two domains: a watershed from central Arizona, Spring Creek and a modeled rectangular landscape obtained by running an existing fluvial landscape evolution model driver in Landlab (Adams et al., 2017). Both landscapes have a drainage area of 36 km^2 and a cell size of 30 m. Rain falls on the landscape and flows downhill, driving overland flow and a hydrograph at every location on the landscape. In this notebook, we track the hydrograph at three

points in the watershed. We recommend that the users review introductory concepts of overland flow and hydrographs before using this notebook and develop familiarity with the term's rainfall intensity and duration, as well as peak discharge, hydrograph time to peak, rising limb and falling limb. our aim is to clearly distinguish the component-based options for designing a storm hydrograph based on the choice of basin, storm intensity, and routing method.

3.1.2 Notebook 1 Interactive Steps

The notebook is designed to run the model several times, each time changing the rainfall characteristics, routing methods or watershed on which flow is routed. Different combinations of model components (or 'model instance') will generate different hydrographs through which the user can explore how different parameters affect hydrograph characteristics. We have provided code to import spatial data linked to the original source of the use of this notebook (Adams et al., 2017) published on HydroShare, so that code can be reproducibly executed with the original ascii text files on a personal computer. In initial runs, the user does not need to change any code, but different scenarios can be developed by switching between test watersheds by changing model parameters such as *basin_flag* to equal "Spring Creek" or to "Square". Table 2 lists the parameters used to obtain the results shown in Figure 2. To generate a storm hydrograph over a modeled time period; approximately 50,000 model timesteps (seconds) could take up to ten minutes of computational run-time (Section 5.0) on existing computer infrastructure (in development for XSEDE; also possible on commercial cloud platforms). We illustrated outputs from the flow routing notebook in Figure 2. The user selects which Landlab component to run: *KinwaveImplicitOverlandflow* or *Overlandflow* components.

Table 2. Parameters used to obtain Spring Creek High Intensity model comparisons between kinematic wave and overland flow model.

Variable	Parameter Description	Value	Dimension
hours	hours of model run time	6	hours
Number_frames	number of frames to plot	6	[-]
n	Manning's roughness coefficient	0.03	s/m^(1/3)
Base_runoff_rate	Base runoff rate	10	mm/h
HigherIntesity_runoff_rate	High intensity runoff rate	20	mm/h
Storm_duration	Storm duration	2	hours
dt	time step for `KinwaveImplicitOverlandFlow`	600	seconds

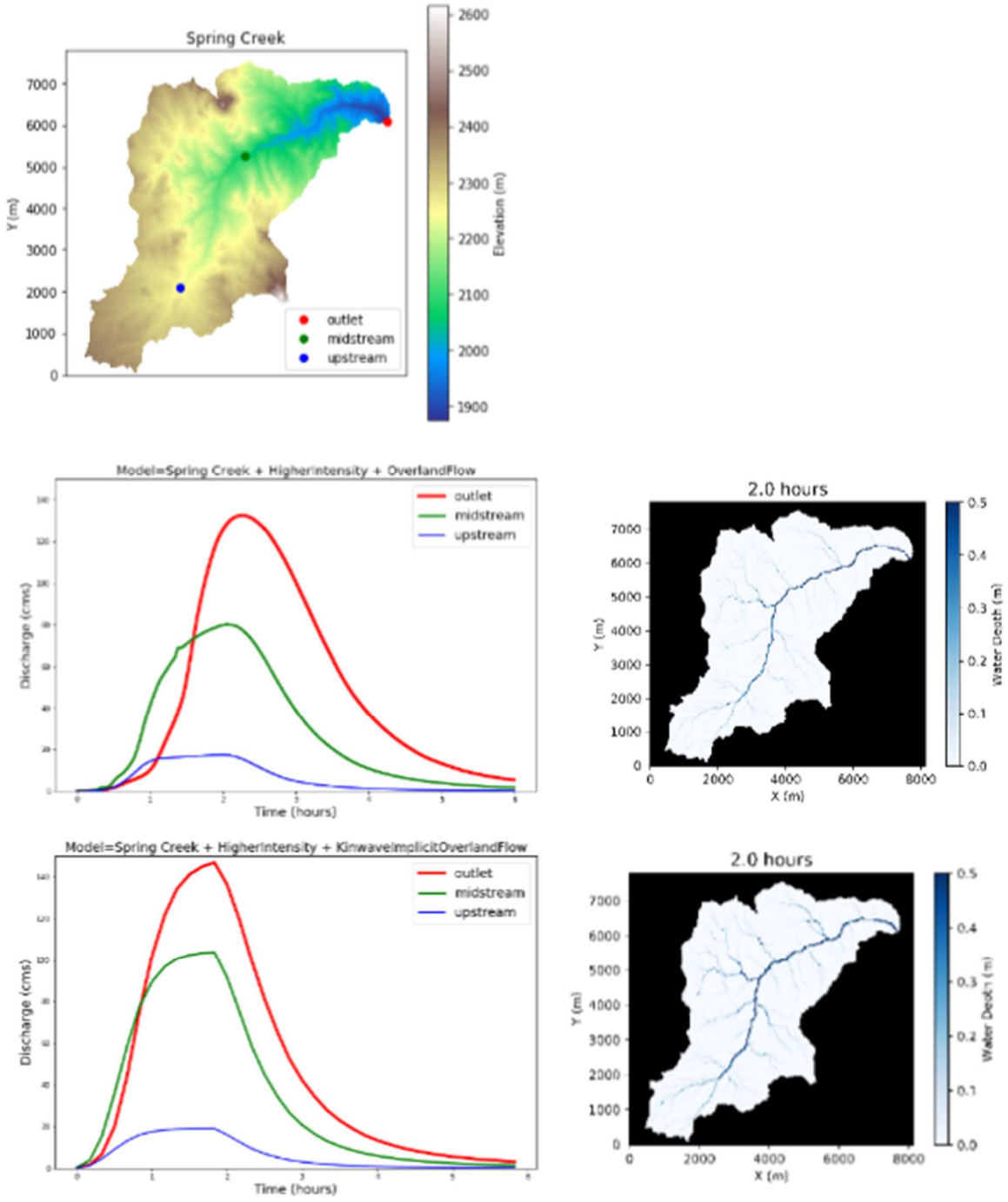


Figure 2. Illustration of flow routing outputs. (a) Elevation map of Spring Creek, central CO with locations (outlet, midstream, upstream) where hydrographs are plotted. (b, d) Hydrographs plotted at three locations shown in (a) driven by the high intensity rainfall option using *KinwaveImplicitOverlandflow* and *Overlandflow* components, respectively. (c, e) Flow depth maps during peak flow for *KinwaveImplicitOverlandflow* and *Overlandflow* components, respectively. Results were produced on HydroShare using the Landlab modeling framework.

3.1.3 Impacts on Numerical Modeling Education.

Using an interactive notebook as a component of the science and engineering curriculum is expected to increase student and faculty access to modeling tools. Rather than relying on software in a computer laboratory or asking students to install new software on their computers, the code can be used in any classroom by every student with access to any computer with a web browser. The example illustrates how model methods and output options can be developed to enhance multifaceted learning experience of the process of interest. In the first notebook there are three main components and various scenarios to explore: two different watersheds, two routing methods and three different storms. Students can simultaneously run scenarios by systematically changing the flags (e.g. *routing_method*, *basin_flag* and *storm_flag*), re-running all code blocks sequentially, and saving the resulting hydrograph plots for each scenario to use in project reporting or homework. The two different flow routing methods show the outcome of including the gradients of fluid pressure and bed elevation, and friction terms of the shallow water equation with different assumptions on hydrographs. Multiple locations for plotting hydrographs in two watersheds will show the role of catchment size and properties. Different excess rainfall intensities are for exploring how increased runoff depth change the hydrograph properties. Advanced students may use the code to build their own visualization, Landlab components, or model optimizations. Because all students can gain hands on experience with the model and code during the classroom instruction, it increases the opportunity and depth of discussions between classmates by providing peer-to-peer learning environment.

3.1.4. Notebook 1 Access

To run this notebook, go to this HydroShare resource (Bandaragoda et al., 2018), click on the blue "Open With" button, select JupyterHub (conceptually this will bring you to block 4 in Figure 1), and execute the first three code blocks of the Welcome page to connect with the cloud computing environment. These steps will certify you are a HydroShare user, download the data and Notebooks from this HydroShare resource to a personal user space in the HydroShare cloud, and print a report of the data that has been downloaded. Click on the hyperlink for *explore_routing_tutorial.ipynb* below the third code block to launch the Notebook described in this section. Alternatively, advanced users (edits are required to remove HydroShare dependencies) can download the Notebook to run on a personal computer with an installed version of Landlab. The Notebook can be directly downloaded (no requirement to become a HydroShare User) at this link: explore_routing_tutorial.ipynb, or viewed on Github in the Landlab organization, tutorials repository, see the *explore_flow_routing* folder.

3.2 Notebook 2: Replicate a landslide model to explore fire impacts on slope instability in a watershed within a regional study

3.2.1. Notebook 2 Overview

Landslides are notoriously challenging to predict (van Westen et al. 2006). A new model developed as a component in Landlab (*LandslideProbability*) offers the ability to predict the probability of shallow landslide initiation at regional scales. Probability of landsliding is

calculated by the infinite-slope stability equation using a Monte Carlo approach by introducing uncertainty to soil, vegetation, and recharge variables. This model was first implemented in a 2,700 km^2 area in the North Cascades National Park Complex (NOCA) of Washington State (Figure 3), where annual probability and return period for shallow landslide initiation was mapped for different soil depth products (Figure 4) (Strauch et al. 2018). Considering the uncertainty of soil depth, root cohesion, and mechanical soil properties, the model predicts 20% to 40% of the area with a landslide return period of 1 in 100 years or less (Figure 4). In comparison to Notebook 1 designed for classroom use, this notebook is designed to replicate model results from Strauch et al. (2018) in Thunder Creek watershed, located within NOCA. It calculates the probability of shallow landslide initiation at a 30-m rectangular grid resolution across the watershed using gridded datasets of landscape characteristics for topography (slope and upslope catchment area), land use and land cover (vegetation type, root cohesion), soil (internal friction angle and transmissivity) and annual maximum daily subsurface flow recharge rate derived from a previously run hydrologic model. All the resources needed for model application are obtained from the existing HydroShare resource from Strauch et al. (2018). Code is provided to import data from the regional NOCA area and create a subset of this data covering Thunder Creek watershed through import of a watershed boundary shapefile. The mean relative wetness and probability of saturated conditions at each grid cell are also calculated in the process of calculating the probability of landsliding. The notebook is designed for exploring the sensitivity of landsliding to environmental conditions that lead to loss of root cohesion, such as a wildfire or timber harvest.

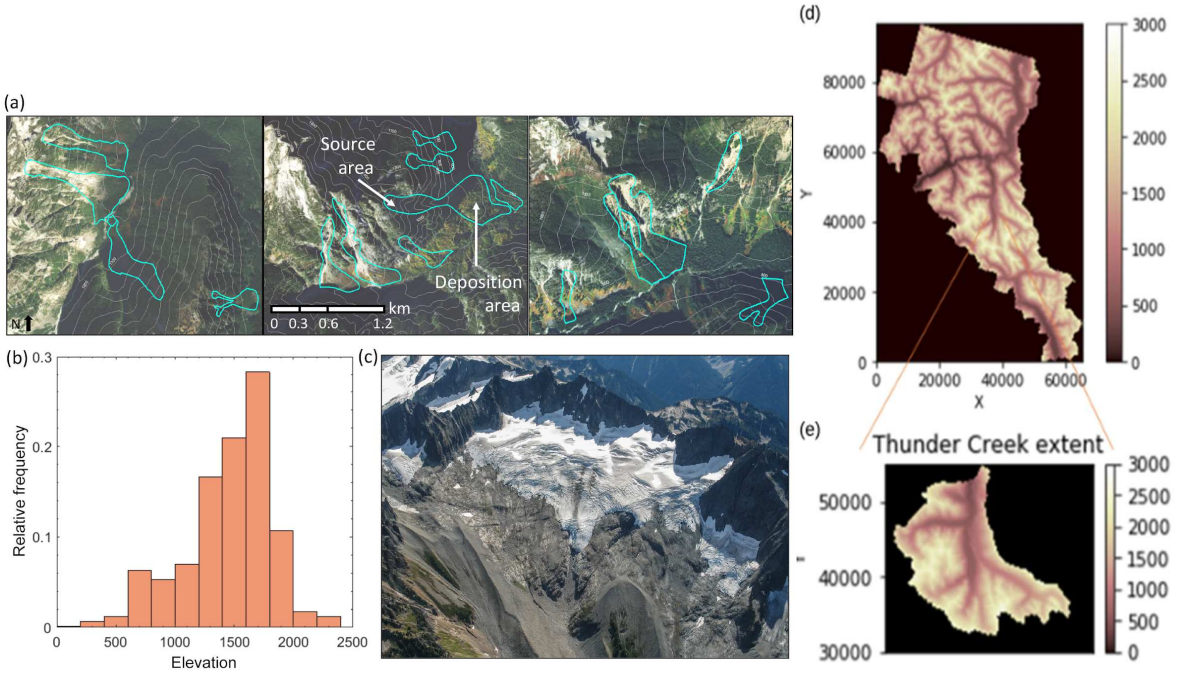


Figure 3. (a) Example debris avalanches (cyan) mapped in three areas within NOCA. Contours are in 100- m intervals. Aerial image source from World Imagery, Esri Inc.; (b) elevation distribution of the relative frequency of mapped debris avalanche source areas ; and (c) High elevation rock and glacier surrounding Spiral Glacier in North Cascades showing a bedrock glacier cirque with thin barren soils and moraine deposits (photo by John Scurlock with

permission), (d) elevation (ft) for NOCA model extent from Strauch et al. (2018), and (e) for the subset for the Thunder Creek extent. (Figures a-c adapted in entirety from Strauch et al., 2018 under CC BY 4.0).

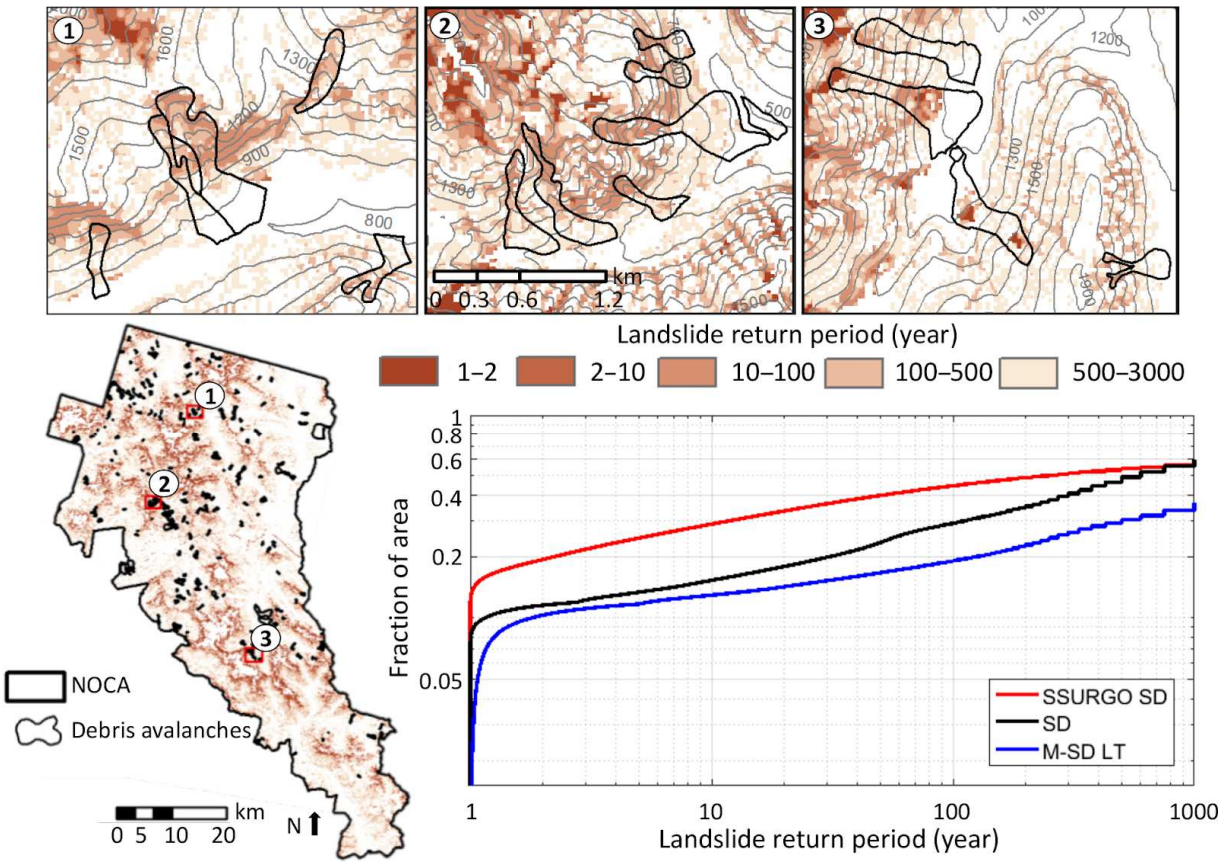


Figure 4. Maps show modeled landslide return periods using Landlab for NOCA overlain with mapped debris avalanches, including zoomed in areas at top for greater detail. The uncertainty of soil depth was characterized from a long-term soil evolution model (M-SD LT). Cumulative distribution of return periods for SSURGO soil depth (SSURGO-SD), modeled soil depth (M-SD), and modeled soil depth considering long-term dynamics (M-SD LT) scenarios, plotted on a log-log scale using the Weibull plotting position. (Figure adapted in entirety from Strauch et al., 2018 under CC BY 4.0).

3.2.2. Notebook 2 Interactive Steps

The Notebook is organized with an introduction (Section 1.0) to the Infinite Slope Factor of Safety Equation, which predicts the ratio of stabilizing to destabilizing forces on a hillslope plane, and the Monte Carlo solution developed to compute probability of landslide initiation. Data Science and Cyberinfrastructure methods are provided in Section 2.0 that describe specifics of accessing existing spatial data, extracting information for the watershed of interest, followed by Landlab Methods for setting model parameters. In the subsection labeled 'Specify Recharge'' our aim is to clearly distinguish the component-based options for studying the impact of assumptions related to recharge and hydrologic forcing on landslide probability. At the end of this section, the number of Monte Carlo iterations is assigned. In

Section 3.0 (Results), the model is executed for Thunder Creek and the results are visualized. Steps for saving results back to HydroShare are listed in Section 4.0.

To support graduate level coursework in hydrologic processes and modeling, we include code blocks that print more explanatory variables and numerical values to verify results are as reported in Strauch et al. (2018). In this demonstration notebook, the user imports necessary Python utilities and libraries and reviews the data needed to execute the landslide model. Code is provided to import data from the regional NOCA area and create a subset of this data covering Thunder Creek watershed through import of a watershed boundary shapefile. One of four recharge options is specified, and the user loads existing mapped landslides to overlay on the landscape to compare with the probabilistic landslide hazard map. The user specifies the number of iterations to use in a Monte Carlo simulation, then runs the *LandslideProbability* component with two cohesion assumptions. The first cohesion assumption is based on existing conditions as described in Strauch et al., (2018). The second cohesion assumption (generating a second model instance) approximates post-fire conditions where root cohesion is reduced by 70%. This represents the reduced root cohesion following a wildfire as existing roots decay following wildfire while new roots begin to regenerate (Sidle, 1992; Istanbulluoglu et al., 2004). Finally, maps are generated to compare the results of the stability analyses and results can be saved back to HydroShare (Figure 5).

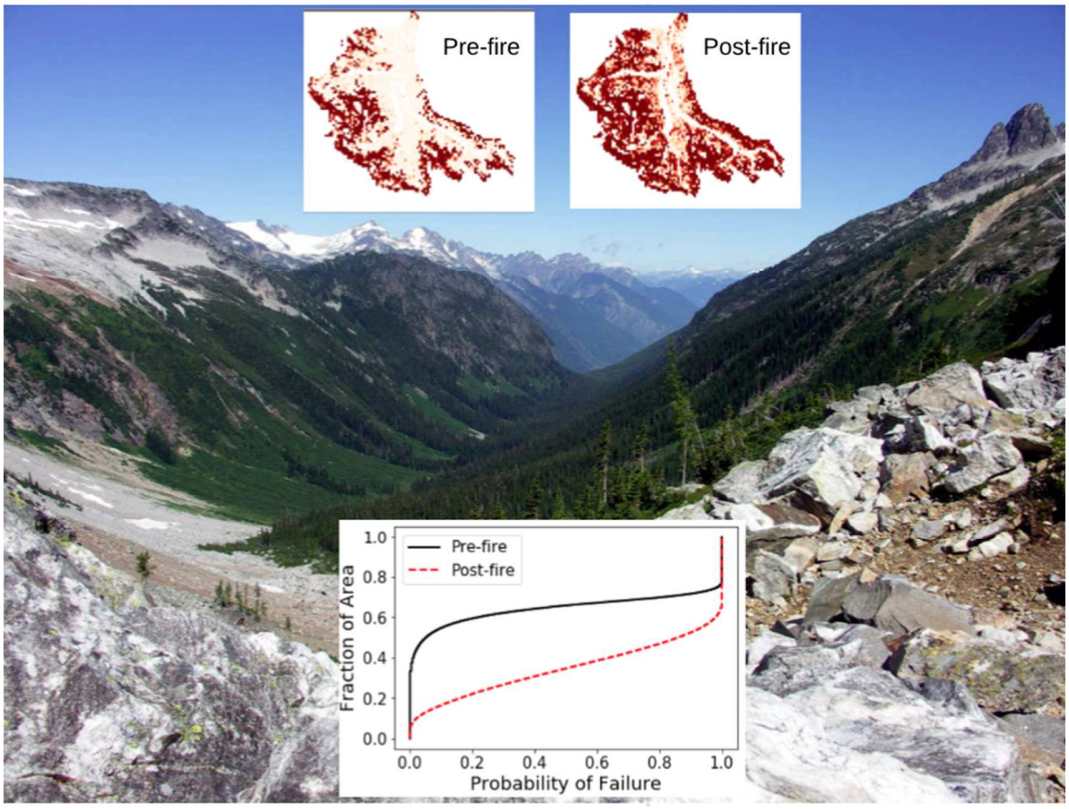


Figure 5. Landslide probability estimates in the Thunder Creek watershed (photo) increase given post-fire root cohesion assumptions (70% less), as compared to the original cohesion assumptions in Strauch et al. (2018). As an example of cyberinfrastructure functionality, the

notebook replicates published findings, as well as tests the parameter function described in the peer-reviewed publication. Inset maps and cumulative distribution plots of the spatial probability of landsliding for pre-fire and post-fire conditions.

Replication of the Strauch et al. (2018) model in Thunder Creek for potential postfire conditions clearly show an increase in annual probability of failure (PF) during when the root cohesion is reduced following wildfire. In the pre-fire simulation, 25% of the landslide is unconditionally unstable PF=1.0, meaning that the soil cannot stand on these slopes. This high annual probability is a conservative estimate and it is largely due to the use of the SSURGO soil depth product in this application. Strauch et al. (2018) discussed how more processed based modeling of soil depth reduce PF to more realistic ranges. With wildfire impact unconditionally unstable regions grew to >40% of the watershed. Before fires, ~40% of the watershed is unconditionally stable PF=0.0. These regions are located in the lower portions of U-shaped pro-glacial valleys (Figure 5). With a vegetation disturbance such as wildfire, this fraction is reduced to <5%, which could lead increased sediment input from the sides of U-shaped valley directly to the valley floor, and result in decline of aquatic habitat quality.

3.2.3. Impacts on replicating scientific findings

This notebook is designed for Earth scientists and stakeholders who are interested in understanding the landslide hazard risk as a probability in space and time. Running this notebook using Landlab leverages the software infrastructure of the Landlab Python toolkit, which standardizes the handling of spatial-temporal data. Executing the notebook on HydroShare allows the ability to store necessary data, deploy the model via a super computer, and see the results, which can be evaluated and shared. Thus, the notebook becomes a one-stop online platform for demonstrating the landslide model and facilitating ease of model augmentation. Current barriers to conducting landslide hazard analysis includes the ability to consider landscape variability, data uncertainty, and hydrological triggering mechanisms over a large spatial scale. This narrative helps reduce the barrier of significant time investment to implementing a complex model by providing the necessary data and code for implementing the Landlab *LandslideProbability* component. As a result, the researcher can see what the model requires and how it runs to produce the results presented in a publication. The notebook can provide an example that can be modified to use in a new study effectively across the nation. Additionally, the barrier to accessing compiled observations and research products is overcome with this notebook, including compiled spatial-temporal visualizations that can be used to communicate results.

3.2.4. Notebook 2 Access

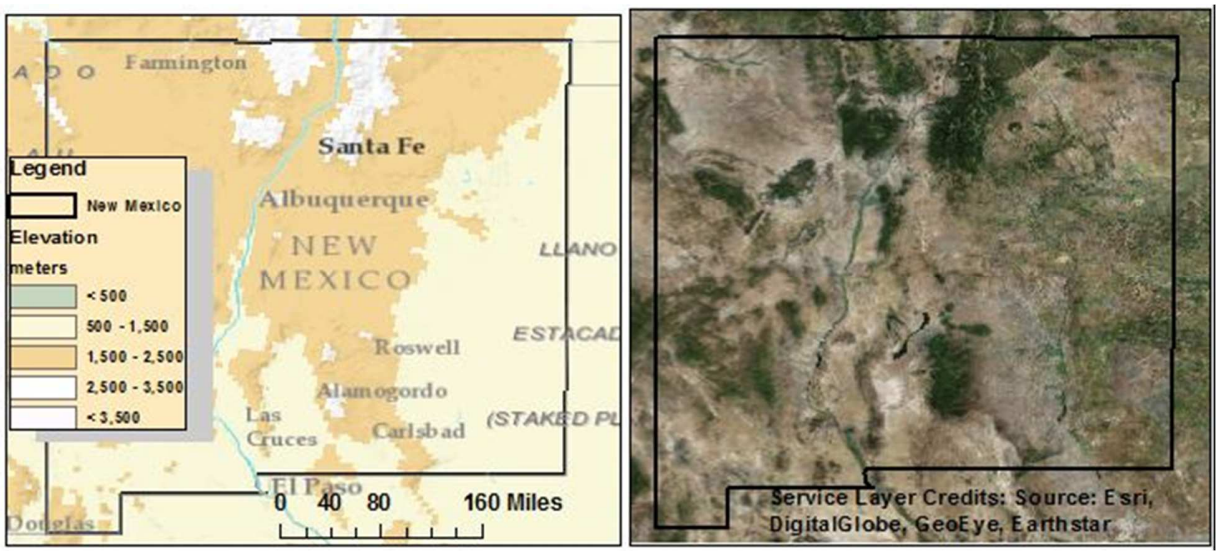
To replicate a published regional Landlab shallow landslide model to explore changes in forest cover at a subcatchment scale, within the NOCA study area, using the Jupyter notebook *replicate_landslide_model_for_fire.ipynb* available on HydroShare, see Bandaragoda et al., (2018).

3.3 Notebook 3. Reuse an Ecohydrology Model with Gridded Hydrometeorology Forcing

3.3.1. Notebook 3 Overview

In semiarid regions climate change and human impact can lead to dramatic changes in the composition and organization of Plant Functional Types (PFTs), such as trees and shrubs, and thus the biomass production of the ecosystem. Ecohydrologic vegetation dynamics models are tools that can be used to explore the role of climatology on the spatial organization of PFTs (Fatichi et al., 2016). In this notebook, we adapt Landlab's ecohydrologic vegetation dynamics model to illustrate how an existing model can be reused by enhancing and developing a workflow at a new location, in our case for studying the role of elevation-dependent precipitation and temperature gradients on PFTs using historical gridded daily weather data from Livneh et al. (2015). Broad elevation bands, low (1200-1700 m), medium (1700-2000 m) and high (2000-2500 m) are developed and the ecohydrology model in Landlab is implemented to simulate the resultant organization of PFTs at each elevation band in the state of New Mexico on hypothetical flat surfaces with a spatially homogenous soil textural properties (Figure 6).

The Landlab ecohydrology mode we used, is based on CATGraSS (Cellular Automaton Tree Grass Shrub Simulator), a discrete time Cellular Automaton (CA) model for spatial evolution of PFTs (Zhou et al., 2013). In CATGraSS each cell in the domain can be occupied by a single PFT: Tree, Shrub, Grass or left unoccupied as bare soil. The model couples local ecohydrologic vegetation dynamics, which simulate biomass production based on local soil moisture and actual evapotranspiration, with spatial processes for plant establishment and mortality controlled by seed dispersal rules, water stress tolerance, and space availability. Trees and shrubs disperse seeds to their neighbors. Grass seeds are assumed to be available everywhere. Establishment of plants in bare cells is determined probabilistically based on water stress of PFTs neighboring the bare cells. Plants with lower water stress have higher probability of establishment. Plant mortality is simulated probabilistically as a result of aging and drought stress. The model is driven by rainfall pulses (observed or generated), solar radiation and temperature. The latter two variables can also be used to prescribe a seasonal potential evapotranspiration input. In Landlab, the model is implemented as a set of interacting components, each describing a different element of the coupled system: *PrecipitationDistribution*, *Radiation*, *PotentialEvapotranspiration*, *SoilMoisture*, *Vegetation* (component for local growth), and *VegCA* (component for cellular automaton rules).



a) b)

Figure 6. Map of elevation bands in New Mexico State (a) used to extract gridded Hydrometeorological forcing data Elevation bins are referred to as: Low elevation (1200-1700 m), mid elevation (1700-2000 m), and high elevation (2000-2500 m). The vegetation patterns from aerial imagery of New Mexico are distinct within these bands (b).

3.3.2. Notebook 3 Interactive Steps and model results

In this notebook, we define the geographic subset (New Mexico) within North America and download gridded hydrometeorologic data from Livneh et al. (2015) for this region. Then, we bin this data into three elevation ranges by considering elevation of centroids of the cells in the gridded dataset and calculate the spatial means of daily precipitation, maximum and minimum temperature for each bin. These data are used to force the ecohydrology model at each elevation bin. The hydrometeorological data handling steps are executed in a separate notebook named *observatory_gridmet_newmexico.ipynb*,located in the folder *ogh_newmexico*, which runs a recently developed Python package for automated retrieval, preprocessing, and visualization of gridded hydrometeorology data products (Phuong et al., 2018). As we described in the Jupyter notebook for this example, we found that the Livneh et al (2015) data had a wet bias in precipitation. This bias is corrected by gathering weather station data (Moore (2011)) that span the range of the elevation bins we used from the Livneh et al. (2015) data. Time series of bias corrected annual precipitation and mean monthly temperature show wetter and cooler conditions as elevation grows (Figure 7). There is a positive trend in annual precipitation from 1950 to 1990, followed by a slight negative trend. In the application of this notebook we suggest the users to explore the model outputs to see if this precipitation trend had any impact on the spatial cover fractions of PFTs. Following the bias correction, the three elevation bins resulted in

climatology's from arid, in the low elevation bin, to semiarid conditions in the high elevation bin according to the aridity index classification (Nash et al., 1999), discussed in relation to model results below. Since the historical data extends only for 64 years (Figure 7), we extended the record to by tiling the daily historical data to facilitate longer vegetation development simulations. The limitation of this approach is that the same climate repeats itself in every 64 years.

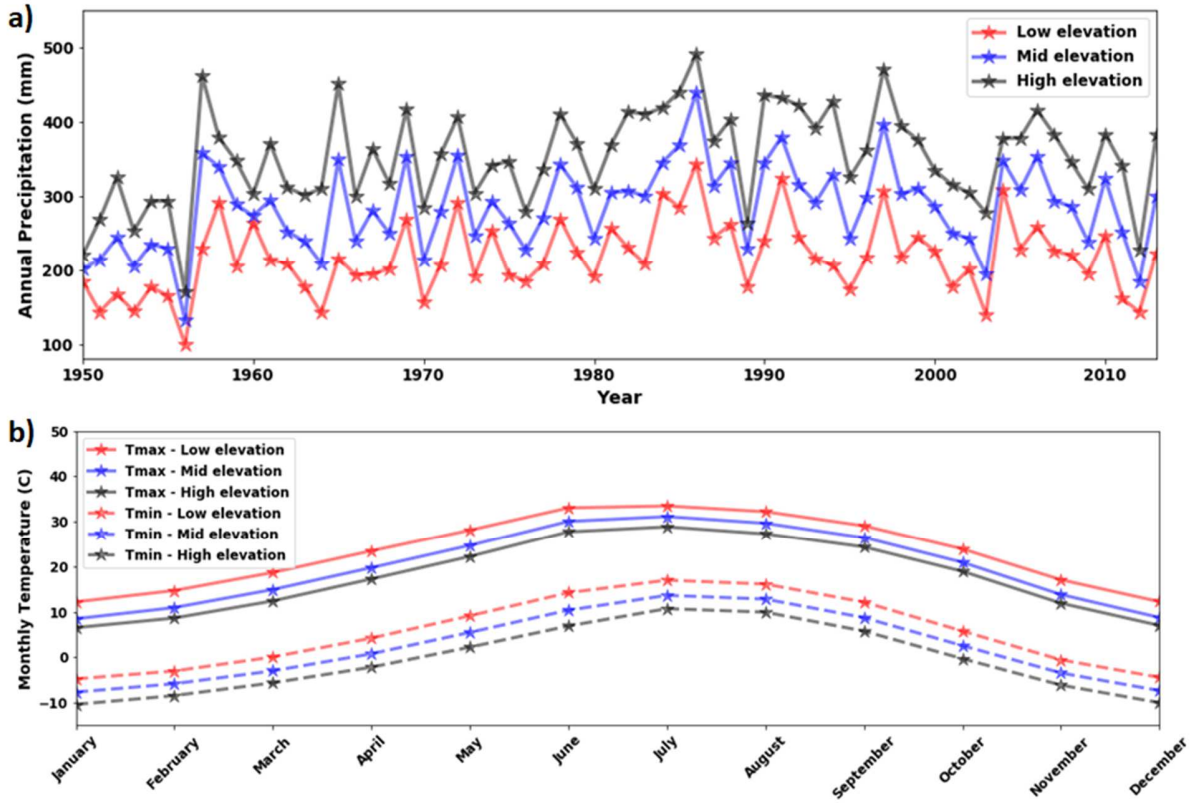


Figure 7. Climate data downloaded and processed from Livneh et al. (2015). a) Annual precipitation plotted with respect to time for each elevation band. b) Mean monthly daily minimum and maximum temperatures for each elevation band.

The notebook presents three model runs to explore the role of elevation-dependent changes in the regional climatology on modeled spatial patterns of PFTs (shrub, grass, tree), and plots the time series of annual areal cover fraction of each PFT that emerge in the domain for a model run time of 1500 years. The Notebook begins with an introduction (Section 1.0) to the Landlab Ecohydrology model, and the Landlab components used to build this model. Data Science and Cyberinfrastructure methods are provided (Section 2.0), followed by Climate Methods (Section 3.0) and Ecohydrology Modeling using Landlab Methods (Section 4.0). Finally, instructions to *Save the results back into HydroShare* (Section 5.0) are given.

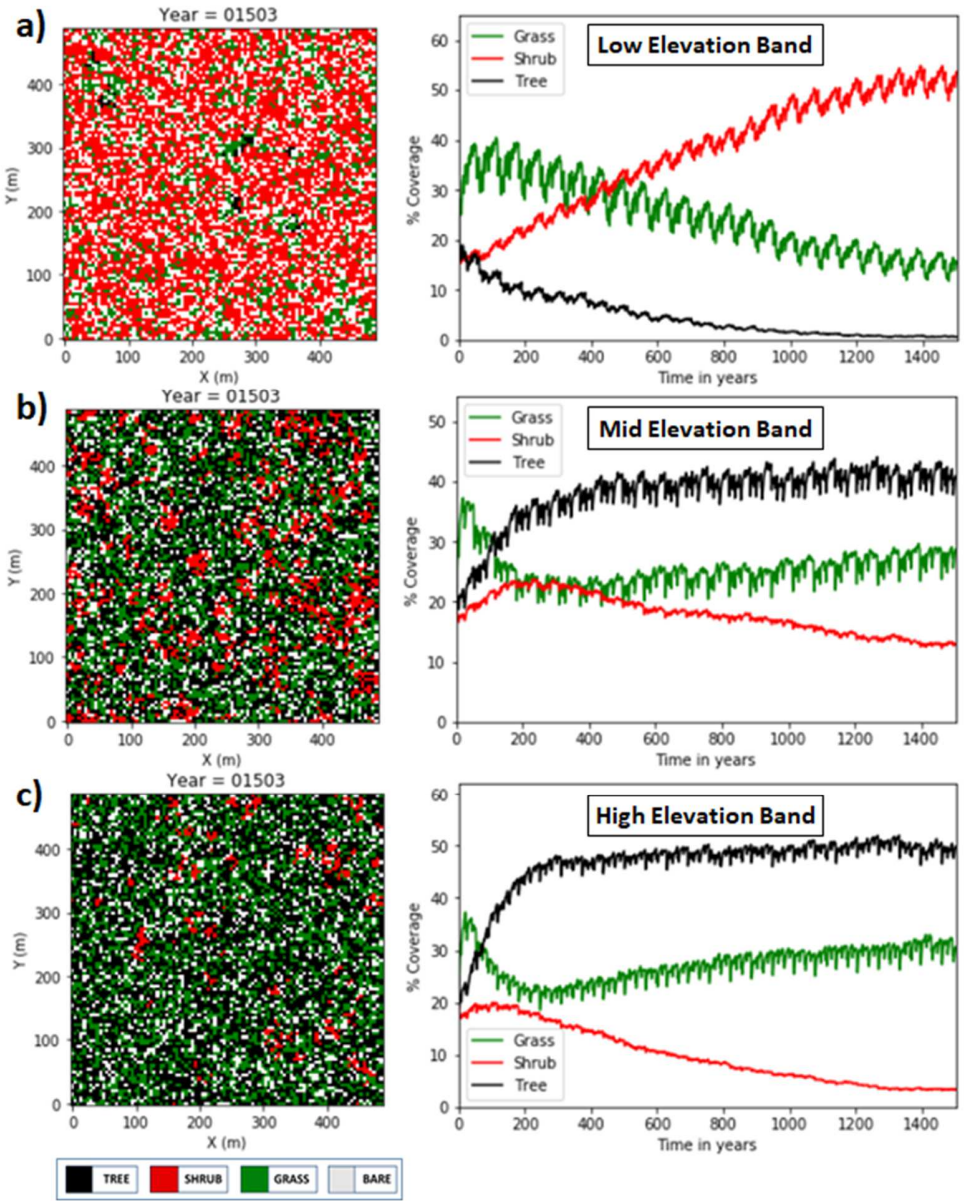


Figure 8. Spatial Organization of PFTs at year 1503 (left column) and Annual areal cover fraction of each PFT plotted with respect to time (right column) for; a) low elevation landscapes (1200 m to 1700 m), b) mid elevation landscapes (1700 m to 2000 m), and c) high elevation landscapes (2000 m to 2500 m).

Starting with a randomly distributed equal fractions of tree, grass, shrub vegetation and bare soil, the model organizes the spatial distribution of PFTs through time. In the low elevation band (MAP = 217mm, PET = 1601mm/y, Aridity Index (AI) = 7.34) the local climate can be considered arid, AI>5 (Nash et al., 1999). Drought-tolerant shrub vegetation outcompetes trees,

leaving a few trees behind, while grass gradually retreats, leading to an ecosystem where shrubs dominate but co-exist with grass as a secondary PFT. The modeled PFT map (Figure 8 a, left) shows pockets of grass clusters within the shrub domain. A few small clusters of trees still exist in very low fraction of the domain. It would be interesting to explore how the grass-shrub interplay shape over longer time using this model. Note that the bell-shaped response of grass, and to an extent, shrubs in this simulation can be attributed to the trends in the precipitation data in the historical period, giving 5% to 10% boost to the areal grass coverage and ~ 5% for shrubs. The repetition of the bell-shaped response is due to the tiling of the historical precipitation and temperature data.

In the mid elevations (MAP = 285mm, PET = 1427mm/y, AI = 5) in the arid to semiarid climate transition (Nash et al., 1999), the conditions are cooler and wetter compared to low elevation band in this example. These conditions provide moisture to sustain enough healthy trees allowing them to outcompete shrubs, as trees can spread seeds to longer distances than shrubs for establishment, to become the primary PFT. Grass grows in empty spaces that are not surrounded by healthy trees or shrubs due to two reasons; 1) the availability of seeds everywhere, 2) lack of PFTs that outcompete them for establishment, as trees and shrubs are competing. This leads to an ecosystem dominated by trees but co-existing with grass as secondary PFT and shrubs as the tertiary PFT (Figure 8 b, right). It will be worthwhile to check whether this ecosystem can sustain the co-existence of the three PFTs for longer periods of time.

At high elevations (MAP = 353mm, PET = 1293mm/y, AI = 3.66), climate is the coolest and wettest among the three elevation bands and fall in the semi-arid category (Nash et al., 1999). Trees dominate shrubs gradually, leading to an ecosystem dominated by trees, while grass retreats gradually and stabilize. Only few small shrub clusters remain after 1500 year. Users can run this model longer to see if shrubs will completely disappear from the ecosystem. In addition to running the notebooks for a longer period of time as discussed above, one can also edit the model inputs by modifying the file *ecohyd_inputs.yaml* (located in the folder *supporting_files*) and explore various hypotheses, for example by changing the soil texture or modifying vegetation parameters to explore how local vegetation dynamics can impact the spatial organization of plants.

3.3.3 Impacts on use and reuse of research models

This notebook is designed for Earth scientists who are interested in understanding the influence of climate on long-term climate-driven changes the spatial vegetation patterns in semi-arid landscapes. The ecohydrologic vegetation dynamics model built in Landlab leverages the framework's flexibility for building numerical models from components and utilities available in its library. In this example, we have demonstrated how to use a Landlab model multiple times on

HydroShare using downloaded gridded meteorological datasets with the OGH library (Phuong et al., 2018).

3.3.4 Notebook 3 Access

To reuse this ecohydrology model using Landlab and HydroShare in our New Mexico example or for another location in the continental United States, use the Jupyter notebook *reuse_ecohydrology_gridhydromet.ipynb* available on HydroShare, see Bandaragoda et al., (2018) or this link.

4 Discussion

Broadly speaking, cyberinfrastructure can be considered a social construct -- components of hardware and software are built by a community of developers based on a perceived need, or by employing user experience research to guide design decisions. When the design of the CI system is a creative and problem-solving endeavor developed with a community committed to using it for their research and education, with feedback and investments of resources -- in which case, the CI be considered Knowledge Infrastructure. We define Knowledge Infrastructure as a web-based system of tools that can be adapted and co-opted to develop technological and sociological solutions to emerging problems of complex systems by efficiently connecting researchers, their data and models, private and public users, and investors committed to long-term maintenance and operations of distributed computing resources. Through the work of developing a description of how others can interact with our Earth surface model results, we learned that the CI outlined in Figure 1 is just one realization of how to synthesize CI components to run Landlab models on HydroShare. We expect that this model will evolve with each research application, model, and user, especially as technology advances and user input improves usability. All users benefit when systematic processes support training for learning new tools and incorporating emerging technology into scientific methods.

There are two main challenges to conducting sophisticated Earth surface model applications, 1) they are computationally and data intensive, and 2) communication of methods and results through traditional peer reviewed journal publications, conference presentations, as well as student-mentor and peer-peer relationships, may not be efficient at ensuring reproducible results. Here we consider “reproducible” to include both the ability to replicate published results (e.g. testable by editors, reviewers, or readers), and to reuse the research products as a baseline for future studies (e.g. accessible code and data). Reproducibility in Earth surface modeling is time and resources expensive; and addressing the challenges above is common across most of computationally intensive sciences requiring research software development. For example, a spatially distributed numerical model application for landslide risk should be reproducible both at the site where model is calibrated and applied in a paper, and the cyberinfrastructure should

provide the flexibility for the same model to be applied at another site just by changing several spatial inputs on the same platform. For another example, an individual researcher may choose a personal *cyberinfrastructure* system (Horsburgh et al., 2016) that they design, develop, and/or inherit from colleagues. Whereas a research collaboration, such as a study by multiple domain scientists and institutions, may require co-design of K*nowledge Infrastructure* to address a broad range of formal and informal processes that support the ongoing development of research products.

We submit for consideration by the Earth surface research community that more attention on designing both personal cyberinfrastructure and shared Knowledge Infrastructure will accelerate our research productivity. The aim is to deploy the latest technologies in such a way as to minimize the researchers' effort to acquire expertise in technologies outside their domain, and to better enable domain scientists to focus their attention on the theoretical underpinnings and development of new process-based understanding of the Earth system. In the current rapidly evolving environment of computer technologies, the community of researchers often needs to keep pace with technological advancements, such as new computational platforms (high performance and cloud computing), open source modeling frameworks, and software paradigms, libraries and tools. We identified five barriers that can be addressed with Knowledge Infrastructure design in research (Section 5.1). We found that these barriers can be lowered by including user input in the system development process (Section 5.2), which we expect to advance science through simultaneously supporting both inductive and deductive learning processes (Section 5.3).

4.1 Defining five common barriers

During our work we notices five common barriers, and sought to lower these for more students and researchers to utilize community resources for numerical modeling:

1. **Unclear processes in conducting open research.** Vocabulary, workflow, and metrics for success are not well understood and standards of practice are at the early stages of development.
2. **Technological requirements for hands-on learning**. Training and workforce development using large datasets and high-performance computing requires expertise beyond the experience of most domain scientists.
3. **Hardware and software requirements for using online infrastructure in workshops and classrooms**. Software installations and model run time on local computers limits the time available to introduce new concepts and tools.
4. **Compiled observations and research products (e.g. model results) are difficult to access.** Data-driven introduction to science concepts is time intensive and there are no best practices for classroom interaction with large datasets and coupled spatial-temporal visualizations of published model results.

5. **Time investment and expertise required to begin using a complex model is too high.** In many collaborations, only one model expert can execute, interact and manipulate the model, which limits building deeper understanding and communicating about implementing new ideas.

4.2 Development based on user input

Cyberinfrastructure can be effective at lowering common barriers if it is designed based on input of users. User information may include cultural, formal, and informal preferences for conducting research and sharing data. Inclusion of user practices to support transfer of knowledge between users, extends cyber-infrastructure to knowledge-infrastructure (KI). For example, design of KI to support the use of the infrastructure to improve communications among users is generally perceived to have the potential to lead to rapid advancements in process and system-level understanding through data analysis and modeling. Scientists and users from multiple research and decision-making communities have shared needs to expand their understanding of processes at specific locations on the Earth surface. For research communities, the focus is always on advancing scientific understanding. For other user communities, such as those applying the latest research to improve data collection, or operating resources based on observational and modeled data, the focus may be on incrementally developing systems to use the gained knowledge to adapt to changing conditions (Mees, 2017; Dilling et al., 2017; Hughes S.A., 2014; Nalau et al., 2015; Baker et al, 2012). Regardless of the academic labeling of the system (CI or KI), users and developers want a simple work experience where they launch a web browser and quickly get to work -- in plain language, using online infrastructure (OI). Regardless of what the purposes of modeling and the background of users might be, the OI should give enough confidence to users to run models, reproduce and reuse model applications, analyze results, and communicate their findings and unique perspectives on the complex system behavior they are investigating.

4.3 Development to advance science

To encourage continuous scientific advancement in the Earth science domain, we advocate that researchers develop their data and models using cyberinfrastructure that enables replication and reuse and consider leveraging open source data and models wherever possible. This approach is ideal for doctoral students such that their data and models would be developed with certain standards and shared in an open source system, that can be replicated, reused and advanced by other investigators. Additionally, code reproducibility rapidly builds the knowledge and skills of other students, thus, shortening the learning curve for modeling, allowing more time to progress research in a domain of science that is supported by using the model, and not distracting from work on a primary research question with data and modeling technical issues. If a user follows an inductive learning narrative, they may use a workflow (Figure 9, top to bottom workflow) that starts with a new idea, and ends with testing a hypothesis, or an inductive learning approach

(Section 5.3.1). If a user follows a deductive learning narrative, they may begin with a pre-existing experiment or toolset, test a hypothesis, and then develop new ideas from what they learn during the tests (Fig.10, bottom to top workflow). Next, we describe how both inductive and deductive narratives are supported.

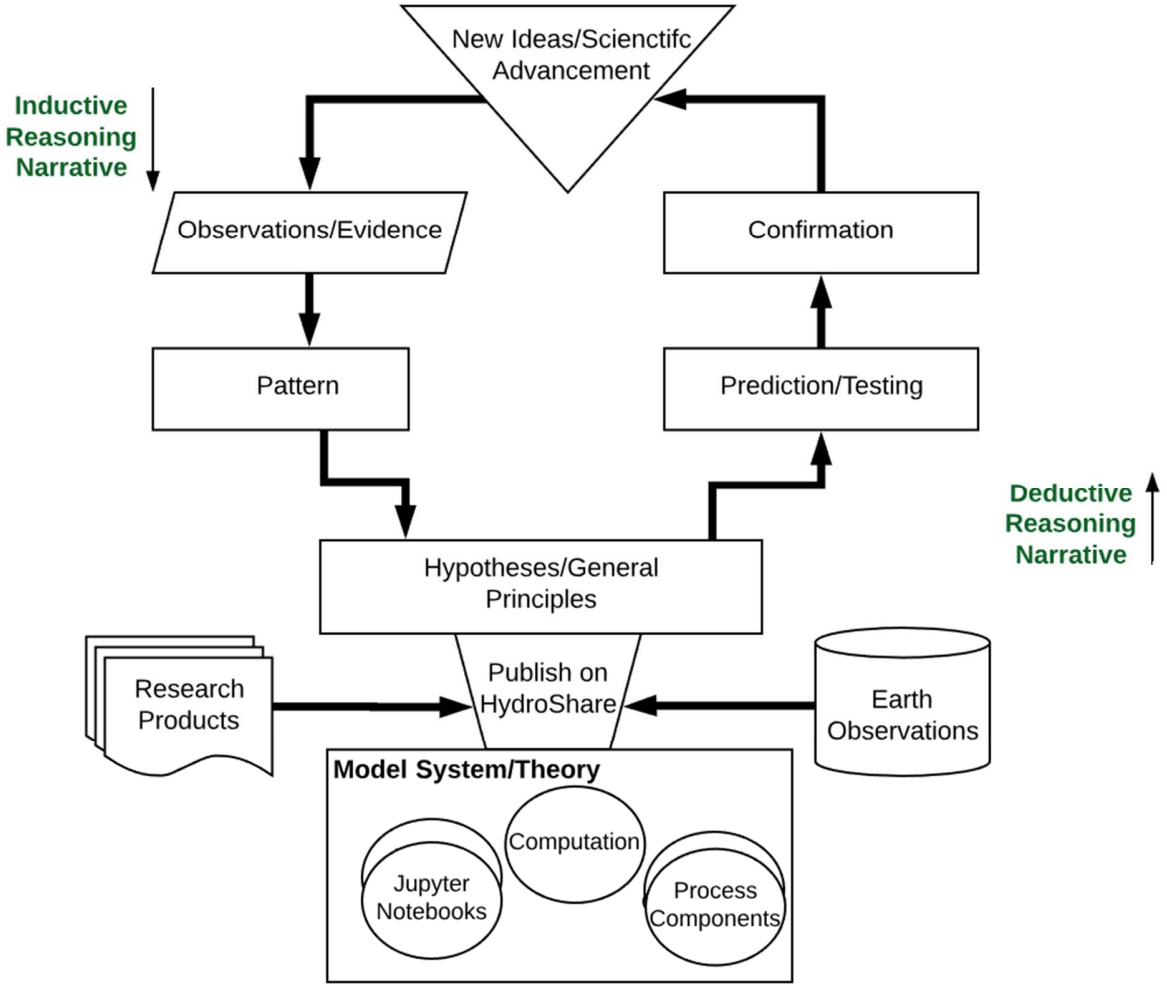


Figure 9. Illustration of community learning and discovery process by code access and utilization among scientists. Key: Triangle = Synthesis/Merge, Circle = Connector, Square = Process, Quadrilateral = Manual or Machine Operation, Cylinder = Database. Inductive processes are supported when, for example, a Landlab user has a new idea for a component, develops the Landlab application, and publishes it on HydroShare (formally with DOI or get a publicly accessible URL), and reviewers can test the experiment with cloud resources. Deductive processes are supported when new Earth observations are published on HydroShare, used to test hypotheses or principals using published models, and results shown to lead to scientific advancements or the development of new ideas.

4.3.1 Inductive learning approach

An inductive learning approach develops evidence and inference by selecting a hypothesized process representation within a system (e.g., landscape), testing that hypothesis, and depending on the outcome develop a refined hypothesis of the process and further design testable numerical and field experiments. This approach is crucial for advancing theoretical concepts for each process and identifying process couplings. Most Earth science models require laborious work to make them suitable for inductive learning. Recent component-based frameworks like Landlab (Hobley et al., 2017) and SUMMA (Clarke et al., 2016)) are developed with the perspective that they can be used for inductive learning and research centered on developing a new idea and making use of other published and tested components in the complex system to test one new idea at a time.

4.3.2 Deductive learning approach

A deductive approach is useful when given a precompiled set of model inputs, outputs, and coupled system of process models; the user or cooperative research group can develop new hypotheses to test given emerging research and new observations. This is a common workflow in science and engineering where a model in published, and the code and data are shared such that when new observations or tools are added onto a published package during continuing research, these addenda are added to an existing library and new ideas for tools, experiments, and data collection emerge.

The preliminary development of Landlab focused on workflow designs where users would begin code development by testing new ideas using published Python scripts to develop process representations of individual Earth system processes. The result is a Landlab environment (Figure 9) with an ecosystem of process components, where users can test new ideas resulting in the development of new components that contribute to a shared and expanding library. While it is common in Earth surface numerical modeling communities to build on and contribute to existing models, the Landlab approach provides a way for new users to begin learning and contributing by developing simple Python scripts that could be executed from a terminal command line. Landlab provides a means for new users to use an inductive learning approach to study one Earth surface process at a time, without having to first master the use of pre-existing complex model and to contribute code to expand processes represented in the model. Running Landlab on HydroShare (Figure 1) provides new users the opportunity to quickly begin exploring Landlab models with minimal software requirements (a web browser and internet connection). Landlab and HydroShare development and research community can continuously

improve and evolve, for example, by implementing an automated updating system that would maintain Landlab version on HydroShare with automated tests the ensure new versions of Landlab continue working with all HydroShare resources that use Landlab.

5 Conclusions

To illustrate how common barriers to Earth surface modeling can be lowered using Knowledge Infrastructure, we have developed three interactive computational narratives using Landlab on HydroShare. Landlab is a recently developed Python-based Earth surface modeling toolkit (Hobley et al., 2017). HydroShare is an hydroinformatics cyberinfrastructure that can be used to store and share hydrologic data and models (Idaszak et al., 2017; www.hydroshare.org). The infrastructure design and methods are illustrated as an interchangeable set of hardware and software components. For our case study we combine an online community repository (HydroShare), modeling framework (Landlab), software environment (dockerized JupyterHub server), and storage (iRODS), with a community approach to advancing scientific progress using Earth surface models. We demonstrate how to use this system in a classroom setting to explore spatio-temporal data, network processes (e.g. hydrologic routing), replicate published results from a complex model in a controlled software environment (e.g. landslide model sensitivity to fire-related parameters), and how to use the same system to reuse flexible components to design a model experiment (e.g. ecohydrology model sensitivity to elevation and climate) that can be used generate new results in any location in the continental United States.

These use cases we present have been designed to illustrate a range of functions and show the benefits of using Knowledge Infrastructure given a range of science topics to address common challenges to using online systems for collaborative numerical modeling. In the past, running distributed hydrology, landslide and ecohydrologic vegetation dynamics Landlab model components required access to a powerful computer, an installation of Python, and an installation of Landlab. Now, any user can log into HydroShare through a live internet browser from any computer/tablet/mobile and run this model, without having to install any software. The user can explore the models further by changing the model parameters, climate forcings, or building their own model with community support. The demonstrated Knowledge Infrastructure, enabled by advanced cyberinfrastructure, is designed to support researchers in more efficiently advancing Earth system sciences.

6 Abbreviations

- Knowledge Infrastructure (KI)
- Cyberinfrastructure (CI)
- Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI)
- Community Surface Dynamics Modeling System (CSDMS)
- Information and communication technologies (ICTs)
- Graphical user interfaces (GUI)
- Personal computer (PC)
- High performance computing (HPC)
- Cloud-based high-performance computing (CHPC)
- Services Oriented Architecture (SOA)
- Open Archives Initiative Object Reuse and Exchange (OAI-ORE) standard
- Digital object identifier (DOI)
- Representative State Transfer (REST)
- Application programming interface (API)
- Structure for Unifying Multiple Modeling Alternatives (SUMMA)
- National Water Model (NWM)
- Observation Data Model 2 (ODM2)
- Extreme Science and Engineering Discovery Environment (XSEDE)
- Resourcing Open Geospatial Education and Research (ROGER) supercomputer
- User experience (UX)

7 Acknowledgments

We acknowledge funding from NSF for HydroShare ACI-1148453 (Tarboton, Bandaragoda), ACI-1148090 (Idaszak), OAC-1664018 (Idaszak), OAC-1664061 (Tarboton, Bandaragoda, Idaszak), and OAC-1664119 (Wang), and CUAHSI EAR-1338606 (Castronova). These development grants supported a team including the Consortium of Universities for the Advancement of Hydrologic Science, Inc (CUAHSI), Brigham Young University, Tufts University, University of Virginia, Purdue University, University of Texas at Austin, San Diego Supercomputing Center, and University of Washington for the development of the HydroShare platform (http://www.hydroshare.org) from 2012-2018. For ROGER supercomputing OAC-1429699 (Wang), and Advanced Cyberinfrastructure, OAC-1443080, OAC-1429699, and OAC-1047916 (Wang), and CyberGIS ACI 1047916 (Wang, Idaszak). For Landlab OAC-1450412 (Istanbulluoglu), OAC-1147454, OAC-1450409 (Tucker, Hobley), OAC-1450338, EAR-1349375, The Oliver Fund of Tulane University (Gasparini, Hutton); for landslide research, CBET-1336725 (Istanbulluoglu); and EAR-1725774 (Barnhart). This project received partial funding from the European Union's Horizon 2020 research and innovation programme under the

Marie Skłodowska-Curie grant agreement No 663830 (Hobley). The authors are grateful for this diversity of contributions. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and developers and do not necessarily reflect the views of the NSF or other funding organizations. All data and code used in this research are available on Github (Landlab and HydroShare organizations) as well as HydroShare (www.hydroshare.org) where readers can Collaborate, Join the public Landlab Group, and view shared public data described in this paper.

8 References

Adams, B.M., Bauman, L.E., Bohnhoff, W.J., Dalbey, K.R., Ebeida, M.S., Eddy, J.P., Eldred, M.S., Hough, P.D., Hu, K.T., Jakeman, J.D., Stephens, J.A., Swiler, L.P., Vigil, D.M., and Wildey, T.M., "Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.0 User's Manual," Sandia Technical Report SAND2014-4633, July 2014. Updated November 2015 (Version 6.3).

Adams, J. M., Gasparini, N. M., Hobley, D. E. J., Tucker, G. E., Hutton, E. W. H., Nudurupati, S. S., & Istanbulluoglu, E. (2017). The Landlab v1.0 OverlandFlow component: a Python tool for computing shallow-water flow across watersheds. *Geoscientific Model Development*, *10*(4), 1645–1663. https://doi.org/10.5194/gmd-10-1645-2017

Almeida, G. A. M. de, Bates, P., Freer, J. E., & Souvignet, M. (2012). Improving the stability of a simple formulation of the shallow water equations for 2-D flood modeling. *Water Resources Research*, *48*(5). https://doi.org/10.1029/2011WR011570

Anders, A. M., Roe, G. H., Montgomery, D. R., & Hallet, B. (2008). Influence of precipitation phase on the form of mountain ranges. *Geology*, *36*(6), 479. https://doi.org/10.1130/G24821A.1

Apache Solr -. (n.d.). Retrieved October 29, 2018, from http://lucene.apache.org/solr/

Atkins, D. E., K Droegemeier, K., Feldman, S. I., Garcia-Molina, H., L Klein, M., Messerschmitt, D., … H Wright, M. (2003). *Atkins Report: Revolutionizing Science and Engineering Through Cyberinfrastructure, Report of the Blue-Ribbon Advisory Panel on Cyberinfrastructure, National Science Foundation, 2003*.

Baker, I., Peterson, A., Brown, G., & McAlpine, C. (2012). Local government response to the impacts of climate change: An evaluation of local climate adaptation plans. *Landscape and Urban Planning*, *107*(2), 127–136. https://doi.org/10.1016/j.landurbplan.2012.05.009

Baldwin, H. (2013, January 18). Tech hotshots: The rise of the UX expert. Retrieved October 19, 2018, from https://www.computerworld.com/article/2493971/app-development/tech-hotshots--the-rise-of-the-ux-expert.html

Bandaragoda, C., Phuong, J., Mooney, S., Stephens, K., Istanbulluoglu, E., Ferguson-Sauder, T., … Idaszak, R. (2018). NSF SI2 Collaborative RAPID Lightning Talk: Building infrastructure to prevent disasters like Hurricane Maria. https://doi.org/10.6084/m9.figshare.6175712.v1

Bandaragoda, C., Tarboton, D., & Maidment, D. (2006). Hydrology's efforts toward the cyberfrontier. *Eos, Transactions American Geophysical Union*, *87*(1), 2–6. https://doi.org/10.1029/2006EO010005

Borgman, C. L., Darch, P. T., Sands, A. E., Pasquetto, I. V., Golshan, M. S., Wallis, J. C., & Traweek, S. (2015). Knowledge infrastructures in science: data, diversity, and digital libraries. *International Journal on Digital Libraries*, *16*(3), 207–227. https://doi.org/10.1007/s00799-015-0157-z

Brooks, K. M. (1996). Do Story Agents Use Rocking Chairs? The Theory and Implementation of One Model for Computational Narrative. In *Proceedings of the Fourth ACM International Conference on Multimedia* (pp. 317–328). New York, NY, USA: ACM. https://doi.org/10.1145/244130.244233

castrona/hydroshare-jupyterhub - Docker Hub. (n.d.). Retrieved October 29, 2018, from https://hub.docker.com/r/castrona/hydroshare-jupyterhub/

Clark, M. P., Kavetski, D., & Fenicia, F. (2011). Pursuing the method of multiple working hypotheses for hydrological modeling. *Water Resources Research*, *47*(9). https://doi.org/10.1029/2010WR009827

Clark, M. P., Nijssen, B., Lundquist, J. D., Kavetski, D., Rupp, D. E., Woods, R. A., … Marks, D. G. (2015). A unified approach for process-based hydrologic modeling: 2. Model implementation and case studies. *Water Resources Research*, *51*(4), 2515–2542. https://doi.org/10.1002/2015WR017200

Clark, M. P., Nijssen, B., Lundquist, J. D., Kavetski, D., Rupp, D. E., Woods, R. A., … Rasmussen, R. M. (2015). A unified approach for process-based hydrologic modeling: 1. Modeling concept. *Water Resources Research*, *51*(4), 2498–2514. https://doi.org/10.1002/2015WR017198

Clark, M. P., Slater, A. G., Rupp, D. E., Woods, R. A., Vrugt, J. A., Gupta, H. V., … Hay, L. E. (2008). Framework for Understanding Structural Errors (FUSE): A modular framework to diagnose differences between hydrological models. *Water Resources Research*, *44*(12). https://doi.org/10.1029/2007WR006735

Dilling, L., Pizzi, E., Berggren, J., Ravikumar, A., & Andersson, K. (2017). Drivers of adaptation: Responses to weather- and climate-related hazards in 60 local governments in the Intermountain Western U.S. *Environment and Planning A*, *49*(11), 2628–2648. https://doi.org/10.1177/0308518X16688686

Ding, T., & Schloss, P. D. (2014). Dynamics and associations of microbial community types across the human body. *Nature*, *509*(7500), 357–360. https://doi.org/10.1038/nature13178

Edwards, P. N., Jackson, S. J., Chalmers, M. K., Bowker, G. C., Borgman, C. L., Ribes, D., … Calvery, S. (n.d.). *Knowledge Infrastructures: Intellectual Frameworks and Research Challenges*.

Flanagan, D. C., Gilley, J. E., & Franti, T. G. (2007). Water Erosion Prediction Project (WEPP): Development History, Model Capabilities, and Future Enhancements. *Transactions of the ASABE*, *50*(5), 1603–1612. https://doi.org/10.13031/2013.23968

Forlizzi, J., & Ford, S. (2000). The Building Blocks of Experience: An Early Framework for Interaction Designers. In *Symposium on Designing Interactive Systems*.

Freeman, P. A., Crawford, D. L., Kim, S., & Munoz, J. L. (2005). Cyberinfrastructure for Science and Engineering: Promises and Challenges. *Proceedings of the IEEE*, *93*(3), 682–691. https://doi.org/10.1109/JPROC.2004.842782

Freire, J., Fuhr, N., & Rauber, A. (2016). Reproducibility of Data-Oriented Experiments in e-Science (Dagstuhl Seminar 16041). https://doi.org/10.4230/dagrep.6.1.108

Gross, A. M., Orosco, R. K., Shen, J. P., Egloff, A. M., Carter, H., Hofree, M., … Ideker, T. (2014). Multi-tiered genomic analysis of head and neck cancer ties TP53 mutation to 3p loss. *Nature Genetics*, *46*(9), 939–943. https://doi.org/10.1038/ng.3051

Han, J., Gasparini, N. M., & Johnson, J. P. L. (2015). Measuring the imprint of orographic rainfall gradients on the morphology of steady-state numerical fluvial landscapes. *Earth Surface Processes and Landforms*, *40*(10), 1334–1350. https://doi.org/10.1002/esp.3723

Hanney, R., & Savin-Baden, M. (2013). The problem of projects: understanding the theoretical underpinnings of project-led PBL. *London Review of Education*, *11*(1). https://doi.org/10.1080/14748460.2012.761816

Hobley, D. E. J., Adams, J. M., Nudurupati, S. S., Hutton, E. W. H., Gasparini, N. M., Istanbulluoglu, E., & Tucker, G. E. (2017). Creative computing with Landlab: an open-source toolkit for building, coupling, and exploring

two-dimensional numerical models of Earth-surface dynamics. *Earth Surface Dynamics*, *5*(1), 21–46. https://doi.org/10.5194/esurf-5-21-2017

Horsburgh, J. S., Aufdenkampe, A. K., Mayorga, E., Lehnert, K. A., Hsu, L., Song, L., … Whitenack, T. (2016). Observations Data Model 2: A community information model for spatially discrete Earth observations. *Environmental Modelling & Software*, *79*, 55–74. https://doi.org/10.1016/j.envsoft.2016.01.010

Horsburgh, J. S., Leonardo, M. E., Abdallah, A. M., & Rosenberg, D. E. (2017). Measuring water use, conservation, and differences by gender using an inexpensive, high frequency metering system. *Environmental Modelling & Software*, *96*, 83–94. https://doi.org/10.1016/j.envsoft.2017.06.035

Horsburgh, J. S., Morsy, M. M., Castronova, A. M., Goodall, J. L., Gan, T., Yi, H., … Tarboton, D. G. (2016). HydroShare: Sharing Diverse Environmental Data Types and Models as Social Objects with Application to the Hydrology Domain. *JAWRA Journal of the American Water Resources Association*, *52*(4), 873–889. https://doi.org/10.1111/1752-1688.12363

Hughes, S. (2014, April). *A Meta-Analysis of Local Climate Change Adaptation Actions | Science Inventory | US EPA*. Presented at the Carolinas Climate Resilience Conference, Charlotte, NC.

Hutton, C., Wagener, T., Freer, J., Han, D., Duffy, C., & Arheimer, B. (2016). Most computational hydrology is not reproducible, so is it really science? *Water Resources Research*, *52*(10), 7548–7555. https://doi.org/10.1002/2016WR019285

Idaszak, R., Tarboton, D. G., Yi, H., Christopherson, L., Stealey, M., Miles, B., … Horsburgh, J. S. (n.d.). HydroShare - A case study of the application of modern software engineering to a large distributed federally-funded scientific software development project. In *Software Engineering for Science* (pp. 219–233). Taylor & Francis CRC Press.

Improving the stability of a simple formulation of the shallow water equations for 2-D flood modeling - Almeida - 2012 - Water Resources Research - Wiley Online Library. (n.d.). Retrieved October 29, 2018, from https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2011WR011570

iRODS. (n.d.). Retrieved October 23, 2018, from https://irods.org/

Jones, A. S., Aanderud, Z. T., Horsburgh, J. S., Eiriksson, D. P., Dastrup, D., Cox, C., … Baker, M. A. (2017). Designing and Implementing a Network for Sensing Water Quality and Hydrology across Mountain to Urban

Transitions. *JAWRA Journal of the American Water Resources Association*, *53*(5), 1095–1120. https://doi.org/10.1111/1752-1688.12557

Jupyter metapackage for installation, docs and chat: jupyter/jupyter. (2018). Python, Project Jupyter. Retrieved from https://github.com/jupyter/jupyter (Original work published 2015)

Jupyter Project. (2016). Jupyter Notebook. Retrieved from http://jupyter.org/

Kadlec, J., StClair, B., Ames, D. P., & Gill, R. A. (2015). WaterML R package for managing ecological experiment data on a CUAHSI HydroServer. *Ecological Informatics*, *28*, 19–28. https://doi.org/10.1016/j.ecoinf.2015.05.002

Kokotsaki, D., Menzies, V., & Wiggins, A. (2016). Project-based learning: A review of the literature. *Improving Schools*, *19*(3), 267–277. https://doi.org/10.1177/1365480216659733

Laflen, J.M. (1997). WEPP-predicting water erosion using a process-based model. *Journal of Soil and Water Conservation*, *v. 52*(2), 96–102.

Laflen, John M., Lane, L. J., & Foster, G. R. (1991). WEPP: A new generation of erosion prediction technology. *Journal of Soil and Water Conservation*, *46*(1), 34–38.

Lagoze, C., Van de Sompel, H., Nelson, M., Warner, S., Sanderson, R., & Johnston, P. (2008). Object Re-Use & Exchange: A Resource-Centric Approach.

Lemos, M. C., Kirchhoff, C. J., & Ramprasad, V. (2012). Narrowing the climate information usability gap. *Nature Climate Change*, *2*, 789–794. https://doi.org/10.1038/nclimate1614

Luo, W., Duffin, K. L., Peronja, E., Stravers, J. A., & Henry, G. M. (2004). A web-based interactive landform simulation model (WILSIM). *Computers & Geosciences*, *30*(3), 215–220. https://doi.org/10.1016/j.cageo.2004.01.001

Luo, W., Pelletier, J., Duffin, K., Ormand, C., Hung, W., Shernoff, D. J., … Furness, W. (2016). Advantages of Computer Simulation in Enhancing Students' Learning About Landform Evolution: A Case Study Using the Grand Canyon. *Journal of Geoscience Education*, *64*(1), 60–73. https://doi.org/10.5408/15-080.1

Mani, I. (2013). Computational Narratology. *The Living Handbook of Narratology*.

Marco A. Janssen, L. N. A. (2008). Towards a Community Framework for Agent-Based Modelling. *Journal of Artificial Societies and Social Simulation*, *11*(26). Retrieved from http://jasss.soc.surrey.ac.uk/11/2/6.html

Mees, H. (2017). Local governments in the driving seat? A comparative analysis of public and private responsibilities for adaptation to climate change in European and North-American cities. *Journal of Environmental Policy & Planning*, *19*(4), 374–390. https://doi.org/10.1080/1523908X.2016.1223540

Mezzanine - The Best Django CMS. (n.d.). Retrieved October 29, 2018, from http://mezzanine.jupo.org/

Mihalevich, B. A., Horsburgh, J. S., & Melcher, A. A. (2017). High-frequency measurements reveal spatial and temporal patterns of dissolved organic matter in an urban water conveyance. *Environmental Monitoring and Assessment*, *189*(11), 593. https://doi.org/10.1007/s10661-017-6310-y

Moore, D. I. (2016). *Meteorology Data from the Sevilleta National Wildlife Refuge, New Mexico (1988- present)*. Environmental Data Initiative. https://doi.org/10.6073/pasta/4d71c09b242602114fb684c843e9d6ac

Moore, R. (2008). Towards a Theory of Digital Preservation | International Journal of Digital Curation. *The International Journal of Digital Curation*, *3*(1). Retrieved from http://www.ijdc.net/article/view/63

Morsy, M. M., Goodall, J. L., Castronova, A. M., Dash, P., Merwade, V., Sadler, J. M., … Tarboton, D. G. (2017). Design of a metadata framework for environmental models with an example hydrologic application in HydroShare. *Environmental Modelling & Software*, *93*, 13–28. https://doi.org/10.1016/j.envsoft.2017.02.028

Nalau, J., Preston, B. L., & Maloney, M. C. (2015). Is adaptation a local responsibility? *Environmental Science & Policy*, *48*, 89–98. https://doi.org/10.1016/j.envsci.2014.12.011

Nash, D., Middleton, N., & Thomas, D. (1999). World Atlas of Desertification. *The Geographical Journal*. https://doi.org/165. 325. 10.2307/3060449

Netzeva, T. I., Worth, A., Aldenberg, T., Benigni, R., Cronin, M. T. D., Gramatica, P., … Yang, C. (2005). Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships. The report and recommendations of ECVAM Workshop 52. *Alternatives to Laboratory Animals: ATLA*, *33*(2), 155–173.

Newman, M. (2003). The Structure and Function of Complex Networks. *SIAM Review*, *45*(2), 167–256. https://doi.org/10.1137/S003614450342480

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., … Yarkoni, T. (2015). Promoting an open research culture. *Science*, *348*(6242), 1422–1425. https://doi.org/10.1126/science.aab2374

Pande, S., & Sivapalan, M. (2017). Progress in socio-hydrology: a meta-analysis of challenges and opportunities. *Wiley Interdisciplinary Reviews: Water*, *4*(4), e1193. https://doi.org/10.1002/wat2.1193

Perez, F., & Granger, B. E. (2015). *Project Jupyter: Computational Narratives as the Engine of Collaborative Data Science*.

Pfister, L., & Kirchner, J. W. (2017). Debates—Hypothesis testing in hydrology: Theory and practice. *Water Resources Research*, *53*(3), 1792–1798. https://doi.org/10.1002/2016WR020116

Ragan-Kelley, B., Walters, W. A., McDonald, D., Riley, J., Granger, B. E., Gonzalez, A., … Caporaso, J. G. (2013). Collaborative cloud-enabled tools allow rapid, reproducible biological insights. *ISME Journal*, *7*(3), 461–464. https://doi.org/10.1038/ismej.2012.123

Shen, H. (2014). Interactive notebooks: Sharing the code. *Nature News*, *515*(7525), 151. https://doi.org/10.1038/515151a

Stocker, M., Paasonen, P., Fiebig, M., Zaidan, M. A., & Hardisty, A. (2018). Curating Scientific Information in Knowledge Infrastructures. *Data Science Journal*, *17*(0), 21. https://doi.org/10.5334/dsj-2018-021

Strauch, R., Istanbulluoglu, E., Nudurupati, S. S., Bandaragoda, C., Gasparini, N. M., & Tucker, G. E. (2018). A hydroclimatological approach to predicting regional landslide probability using Landlab. *Earth Surface Dynamics*, *6*(1), 49–75. https://doi.org/10.5194/esurf-6-49-2018

Tarboton, D. & The HydroShare Team. (2018, August 23). Collaborative Research: SI2-SSI: Cyberinfrastructure for Advancing Hydrologic Knowledge through Collaborative Integration of Data Science, Modeling and Analysis. National Science Foundation. Retrieved from https://www.nsf.gov/awardsearch/showAward?AWD_ID=1664061&HistoricalAwards=false

Tarboton, D., Bandaragoda, C., & Brazil, L. (2018, January 1). About. Retrieved October 23, 2018, from https://help.hydroshare.org/about-hydroshare/

Tarboton, D., Idaszak, R., Horsburgh, J., Heard, J., Ames, D., Goodall, J., … Merwade, V. (2014). A Resource Centric Approach For Advancing Collaboration Through Hydrologic Data And Model Sharing. *International Conference on Hydroinformatics*. Retrieved from https://academicworks.cuny.edu/cc_conf_hic/314

Tarboton, David G., Idaszak, R., Horsburgh, J. S., Heard, J., Ames, D., Goodall, J. L., … Maidment, D. (2014). Hydro share: Advancing collaboration through hydrologic data and model sharing. Presented at the 7th International Congress on Environmental Modelling and Software, iEMSs 2014. Retrieved from https://uncch.pure.elsevier.com/en/publications/hydro-share-advancing-collaboration-through-hydrologic-data-and-m

Tarboton, D.G. (n.d.). Terrain Analysis Using Digital Elevation Models (TauDEM). Utah Water Research Laboratory, Utah State University. Retrieved from http://hydrology.usu.edu/taudem

Tesfa, T. K., Tarboton, D. G., Watson, D. W., Schreuders, K. A. T., Baker, M. E., & Wallace, R. M. (2011). Extraction of hydrological proximity measures from DEMs using parallel processing. *Environmental Modelling & Software*, *26*(12), 1696–1709. https://doi.org/10.1016/j.envsoft.2011.07.018

The Web framework for perfectionists with deadlines | Django. (n.d.). Retrieved October 29, 2018, from https://www.djangoproject.com/

Tucker, G. E., & Hancock, G. R. (2010). Modelling landscape evolution. *Earth Surface Processes and Landforms*, *35*(1), 28–50. https://doi.org/10.1002/esp.1952

Wang, S. (2010). A CyberGIS Framework for the Synthesis of Cyberinfrastructure, GIS, and Spatial Analysis. *Annals of the Association of American Geographers*, *100*(3), 535–557. https://doi.org/10.1080/00045601003791243

Wang, S. (2016). CyberGIS and spatial data science. *GeoJournal*, *81*(6), 965–968. https://doi.org/10.1007/s10708-016-9740-0

Wang, S., & Goodchild, M. F. (Eds.). (2019). *CyberGIS for Geospatial Discovery and Innovation*. Springer Netherlands. Retrieved from //www.springer.com/us/book/9789402415292

Yetemen, O., Istanbulluoglu, E., Flores-Cervantes, J. H., Vivoni, E. R., & Bras, R. L. (2015). Ecohydrologic role of solar radiation on landscape evolution. *Water Resources Research*, *51*(2), 1127–1157. https://doi.org/10.1002/2014WR016169

Yi, H., Idaszak, R., Stealey, M., Calloway, C., Couch, A. L., & Tarboton, D. G. (2018). Advancing distributed data management for the HydroShare hydrologic information system. *Environmental Modelling & Software*, *102*, 233–240. https://doi.org/10.1016/j.envsoft.2017.12.008

Yin, D., Liu, Y., Padmanabhan, A., Terstriep, J., Rush, J., & Wang, S. (2017). A CyberGIS-Jupyter Framework for Geospatial Analytics at Scale. In *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact* (pp. 18:1–18:8). New Orleans, LA, USA: ACM. https://doi.org/10.1145/3093338.3093378

25 Highest Paying Jobs In Demand - Glassdoor Blog. (n.d.). Retrieved October 23, 2018, from https://www.glassdoor.com/blog/highest-paying-jobs-demand/

9 Figures and Tables

1. **Figure 1.** Illustration of six basic elements for a knowledge cyberinfrastructure for interactive community modeling and exploration. Research software communities maintain support of operations between Docker Containers and software environment. Domain science communities maintain support for version control and user communications specific to modeling frameworks.
2. **Table 1.** Three study problems were designed with a focus to 1) explore, 2) replication, and 3) reuse research. Computational narratives demonstrate how to use Knowledge Infrastructure for computation and visualization of Earth surface models where the user interacts with the infrastructure to develop their own story.
3. **Table 2.** Parameters used to obtain Spring Creek High Intensity model comparisons between kinematic wave and overland flow model.
4. **Figure 2.** Illustration of flow routing outputs. (a) Elevation map of Spring Creek, central CO with locations (outlet, midstream, upstream) where hydrographs are plotted. (b, d) Hydrographs plotted at three locations shown in (a) driven by the high intensity rainfall option using *KinwaveImplicitOverlandflow* and *Overlandflow* components, respectively. (c, e) Flow depth maps during peak flow for *KinwaveImplicitOverlandflow* and *Overlandflow* components, respectively. Results were produced on HydroShare using the Landlab modeling framework.
5. **Figure 3.** (a) Example debris avalanches (cyan) mapped in three areas within NOCA. Contours are in 100- m intervals. Aerial image source from World Imagery, Esri Inc.; (b) elevation distribution of the relative frequency of mapped debris avalanche source areas ; and (c) High elevation rock and glacier surrounding Spiral Glacier in North Cascades showing a bedrock glacier cirque with thin barren soils and moraine deposits (photo by John Scurlock with permission), (d) elevation (ft) for NOCA model extent from Strauch et al. (2018), and (e) for the subset for the Thunder Creek extent. (Figures a-c adapted in entirety from Strauch et al., 2018 under CC BY 4.0).
6. **Figure 4.** Maps show modeled landslide return periods using Landlab for NOCA overlain with mapped debris avalanches, including zoomed in areas at top for greater detail. The uncertainty of soil depth was characterized from a long-term soil evolution model (M-SD LT). Cumulative distribution of return periods for SSURGO soil depth (SSURGO-SD), modeled soil depth (M-SD), and modeled soil depth considering long-term dynamics (M-SD LT) scenarios, plotted on a log-log scale using the Weibull plotting position. (Figure adapted in entirety from Strauch et al., 2018 under CC BY 4.0).
7. **Figure 5.** Landslide probability estimates in the Thunder Creek watershed (photo) increase given post-fire root cohesion assumptions (70% less), as compared to the original cohesion assumptions in Strauch et al. (2018). As an example of cyberinfrastructure functionality, the notebook replicates published findings, as well as tests the parameter function described in the peer-reviewed publication. Inset maps

and cumulative distribution plots of the spatial probability of landsliding for pre-fire and post-fire conditions.

8. **Figure 6.** Map of elevation bands in New Mexico State (a) used to extract gridded Hydrometeorological forcing data Elevation bins are referred to as: Low elevation (1200-1700 m), mid elevation (1700-2000 m), and high elevation (2000-2500 m). The vegetation patterns from aerial imagery of New Mexico are distinct within these bands (b).

9. **Figure 7.** Climate data downloaded and processed from Livneh et al. (2015). a) Annual precipitation plotted with respect to time for each elevation band. b) Mean monthly daily minimum and maximum temperatures for each elevation band.
10. **Figure 8.** Spatial Organization of PFTs at year 1503 (left column) and Annual areal cover fraction of each PFT plotted with respect to time (right column) for; a) low elevation landscapes (1200 m to 1700 m), b) mid elevation landscapes (1700 m to 2000 m), and c) high elevation landscapes (2000 m to 2500 m).
11. **Figure 9**. Illustration of community learning and discovery process by code access and utilization among scientists. Key: Triangle = Synthesis/Merge, Circle = Connector, Square = Process, Quadrilateral = Manual or Machine Operation, Cylinder = Database. Inductive processes are supported when, for example, a Landlab user has a new idea for a component, develops the Landlab application, and publishes it on HydroShare (formally with DOI or get a publicly accessible URL), and reviewers can test the experiment with cloud resources. Deductive processes are supported when new Earth observations are published on HydroShare, used to test hypotheses or principals using published models, and results shown to lead to scientific advancements or the development of new ideas.