GSFLOW–GRASS v1.0.0: GIS-enabled hydrologic modeling of coupled groundwater–surface-water systems

G.-H. Crystal Ng[1,2], Andrew D. Wickert[1,2], Lauren D. Somers[3], Leila Saberi[1], Collin Cronkite-Ratcliff[4], Richard G. Niswonger[5], and Jeffrey M. McKenzie[3]

[1]Department of Earth Sciences, University of Minnesota, Minneapolis, Minnesota, USA
[2]St. Anthony Falls Laboratory, University of Minnesota, Minneapolis, Minnesota, USA
[3]Department of Earth and Planetary Sciences, McGill University, Montreal, Quebec, Canada
[4]Geology, Minerals, Energy and Geophysics Science Center, United States Geological Survey, Menlo Park, California USA
[5]Earth Systems Modeling Branch, United States Geological Survey, Menlo Park, California, USA

Correspondence to: G.-H. Crystal Ng (gcng@umn.edu)

Abstract.

The importance of water moving between the atmosphere and aquifers has led to efforts to develop and maintain coupled models of surface water and groundwater. However, developing inputs to these models is usually time-consuming and requires extensive knowledge of software engineering, often prohibiting their use by many researchers and water managers, and thus

5   reducing these models' potential to promote science-driven decision-making in an era of global change and increasing water-resource stress. In response to this need, we have developed GSFLOW–GRASS, a bundled set of open-source tools that develops inputs for, executes, and graphically displays the results of GSFLOW, the U.S. Geological Survey's coupled groundwater and surface-water flow model. In order to create a robust tool that can be widely implemented over diverse hydro(geo)logic settings, we built a series of GRASS GIS extensions that automatically discretizes a topological surface-water flow network

10   that is linked with an underlying gridded groundwater domain. As inputs, GSFLOW–GRASS requires at a minimum a digital elevation model, a precipitation and temperature record, and estimates of channel parameters and hydraulic conductivity. We demonstrate the broad applicability of the toolbox by successfully testing it in environments with varying degrees of drainage integration, landscape relief, and grid resolution, as well as the presence of irregular coastal boundaries. These examples also show how GSFLOW–GRASS can be implemented to examine the role of groundwater–surface-water interactions in a diverse

15   range of water resources and land management applications.

1 Introduction

Predicting and understanding the hydrologic impacts of climate, land use, and other natural and anthropogenic change is a scientific endeavor that is increasingly necessary to manage water resources. Addressing this need requires streamlined access to models that integrate surface and subsurface processes across a watershed. This integrated approach is required because traditional hydrologic models that focus only on a single component within a watershed cannot properly predict the effects of changing conditions and feedbacks across their boundaries. The widespread use of integrated models is stymied, however, by labor-intensive requirements for creating consistent sets of extensive model inputs, including the challenges of generating computationally robust surface and sub-surface model domains.

Driven by the growing recognition of tightly coupled groundwater and surface water dynamics and the need to evaluate and manage the two as a single resource (Winter et al., 1998), the U.S. Geological Survey (USGS) developed and released GSFLOW. This integrated hydrologic model couples the groundwater flow model MODFLOW with the rainfall–runoff model PRMS (Precipitation Runoff Modeling System) (Markstrom et al., 2008). Both MODFLOW (Harbaugh, 2005; Niswonger et al., 2011) and PRMS (Leavesley et al., 1983; Markstrom et al., 2015) are popular models with significant user bases. GSFLOW has been previously applied to various watersheds in the US, for example in California (Essaid and Hill, 2014), Wisconsin (Hunt et al., 2013), Pennsylvania (Galeone et al., 2016), and Oregon (Surfleet and Tullos, 2013; Gannett et al., 2017), as well as to applications outside of the US (e.g., Hassan et al., 2014; Tian et al., 2015).

The process of implementing GSFLOW includes many hurdles that require significant time and computational knowledge to overcome. GSFLOW is not "fully integrated" in the sense that it does not simultaneously solve surface and subsurface flow equations; instead it consists of an iterative coupling between MODFLOW and PRMS that requires nearly all the individual input files for each of the two original models as well as an additional GSFLOW-specific linkage file. While a fully integrated model may have all the input information streamlined into a small number of internally consistent and efficiently organized files, to run GSFLOW, the user bears the burden of generating a multitude of diversely formatted ASCII files and ensuring that they contain inputs that correctly correspond with each other and can produce convergent coupled simulations. Freely available USGS GUIs – ModelMuse (Winston, 2009) and the PRMS GUI (Markstrom et al., 2015) – and proprietary GUIs (mostly for MODFLOW) can help users separately develop inputs to the two individual base models, but they do not offer support for creating the GSFLOW linkage file. The company Earthfx (http://www.earthfx.com/) provides full GSFLOW support as part of their "VIEWLOG" package, designed for the environmental consulting industry. More openly accessible software endeavors have also improved the usability of integrated hydrologic models (Bhatt et al., 2014; Tian et al., 2016; Gardner et al., 2018), including the USGS's new input data processing tool for GSFLOW (Gardner et al., 2018), but the community still lacks a free and complete package spanning pre- to post-processing for heterogeneous surface and subsurface domains. This gap motivates our present work, which we anticipate will enable more widespread hydrologic modeling.

Our overarching goal is to develop a bundled package – "GSFLOW–GRASS" – to handle the complexity of the coupled GSFLOW model, thus tackling the grand challenge of accessibility plaguing many integrated modeling systems. We develop an integrated toolbox featuring fully automated, robust, and open-source codes that cover the entire model implementation

process within a consistent and efficient framework, from building topologically linked hydrologic domains and assembling model input parameters to visualizing model outputs. Our use of only free and open-source programming languages and software is a key feature of the toolbox's accessibility. Python scripts generate model input files and model output graphics, and extensions using the open-source GRASS GIS platform build topographically defined sub-watersheds linked to subsurface grid cells. Open-source software facilitates implementation of GSFLOW–GRASS by diverse academic, government, and individual entities, enables further community development of GSFLOW–GRASS, and aligns with the USGS's goal to make its resources publicly accessible.

Developing a fully automated toolbox that can be readily executed for diverse physical settings raises the key technical obstacle of how to robustly build stream networks and sub-basins linked to subsurface computational domains without labor-intensive user intervention. Whereas overland flow routing and the calculation of drainage basins from topography are standard GIS capabilities, our tool improves upon these by automatically building topologically structured vectorized drainage networks without manual corrections using a least-cost path approach (Metz et al., 2011), while also including information on adjacency and routing pathways through the network that is required by integrated hydrologic models. The main technical advancement of GSFLOW–GRASS is the development of streamlined GRASS GIS extensions that have passed a diverse range of stress tests, including steep to low-relief topographies, large and intricate to small and simple drainage systems, incomplete to full topographic drainage integration, and inland to coastal watersheds. These new capabilities enable rapid, automated delineation of surface-water drainage networks linked to subsurface domains across any generalized landscape and computationally feasible resolution within the range of scales permissible by GSFLOW. By doing this all within a framework that also includes open-source model input and post-processing tools, GSFLOW–GRASS presents a solution toward more accessible integrated hydrologic modeling.

The release of "GSFLOW–GRASS" coincides with the USGS's new development of GSFLOW–Arcpy, an input data processing tool similarly aimed at facilitating the use of GSFLOW (Gardner et al., 2018). The two software packages solve the problem of generating linked surface water and groundwater model domains using complementary technical formulations and approaches. The major distinction is GSFLOW–GRASS's use of a topographically determined surface water discretization, while GSFLOW–Arcpy employs a regular rectangular domain. Each domain type has its strengths: regular grids allow flexible representation of spatially heterogeneous properties, while topographically based units can efficiently cover steep and complex terrain. The availability of these two packages will facilitate rigorous testing of the comparative merits of the different domain types, which remains an open question in hydrologic modeling (see model inter-comparison tests that include representatives of each (e.g., Reed et al., 2004; Maxwell et al., 2014)). Together, these software toolboxes can also reach more users with different needs and resources. GSFLOW–Arcpy's implementation is well-suited for ArcGIS users who can separately leverage existing MODFLOW tools and are primarily seeking to create surface-water input components for GSFLOW. GSFLOW–GRASS aims to provide a fully automated toolbox that can be accessible to new hydrologic modelers who lack access to commercial ArcGIS software and would benefit from a complete suite of pre- and post-processing tools to execute GSFLOW. Offering different model implementation options affords greater flexibility depending on a user's software preferences, data formats, and application requirements, which we anticipate will help grow a diverse community of integrated hydrologic model users.

2 Background

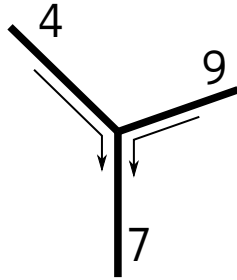
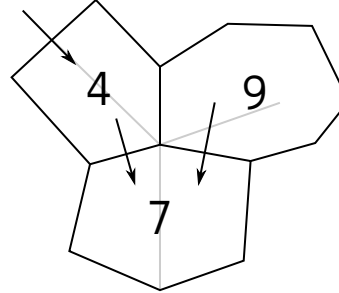
2.1 GSFLOW

GSFLOW simulates spatially distributed surface to subsurface water flow in a watershed using modified model codes from PRMS and MODFLOW. It is designed for simulations of watersheds with areas of a few square kilometers to several thousand square kilometers (Markstrom et al., 2008, p. 2). Although GSFLOW can run in modes equivalent to the stand-alone PRMS-IV model and the stand-alone MODFLOW model, only the "integrated" version is described here. Near-surface watershed processes within the shallow "soil zone," including evapotranspiration, infiltration, runoff, and interflow, are represented by the PRMS sub-component of GSFLOW. Groundwater flow below the "soil zone," including vertical soil water movement in the deeper unsaturated zone and saturated flow through horizontal aquifer layers, is represented by the MODFLOW sub-component. Streamflow and exchange between streams and underlying groundwater systems are also represented by the MODFLOW sub-component. We describe here the key features of GSFLOW in order to guide new users in implementing it and interpreting its results; Markstrom et al. (2008) document the full details of the model.

2.1.1 Domain discretization

GSFLOW adopts a hybrid spatial domain discretization approach (Figure 1) to establish its computational units. Stream segments are links in a river network that are used in both the PRMS and MODFLOW sub-components of GSFLOW (Figure 1A). Horizontally, the PRMS sub-component uses hydrological response units (HRUs) of any shape as its fundamental discretized unit (Figure 1B). These are used for calculations of the upper soil zone and the part of the surface not covered by the stream network. The MODFLOW sub-component uses rectangular grid cells for the deeper subsurface (Figure 1C) and to further discretize the stream network into reaches (Figure 1D). Establishing reaches as the fundamental unit of computation for the stream network instead of segments makes it possible to resolve fine spatial resolution groundwater-surface exchanges. Like MODFLOW grid cells, HRUs can be set to rectangles, but they are also commonly defined topologically to correspond to sub-basins, as they are in our approach (Figure 1). Model intercomparison projects have included both representatives that use gridded domains and those that use irregular domains (Reed et al., 2004; Maxwell et al., 2014). In general, gridded domains are easier to construct and extend readily to parallelized computational systems, and they allow flexible spatial specification of soil and land-cover heterogeneity. In contrast, ungridded domains, such as the triangulated irregular networks (TINs) used in models including tRIBS (Vivoni et al., 2004) and PIHM (Qu and Duffy, 2007), can conform more efficiently to complex terrain. In the case of PIHM (Qu and Duffy, 2007), TINs were also implemented for better water balance performance through the mass-conserving finite volume method (LeVeque, 2002); further, nested TINs can provide efficient solutions when higher resolution is desired for certain target areas (Wang et al., 2018). Other hydrological models with ungridded domains use topographically defined sub-basins as efficient computational units, including SWAT (Arnold and Fohrer, 2005), SAC-SMA (Ajami et al., 2004), HEC-HMS (Feldman, 2000), and TOPNET (Bandaragoda et al., 2004).

Vertically, the PRMS sub-component of GSFLOW is discretized into conceptual shallow soil zone reservoirs, which do not correspond directly to physical locations within the soil column but are instead based on user-specified conceptual thresholds.

A. Segments (MODFLOW)
Links in river network

B. HRUs (PRMS)
Sub-catchments

C. Grid (MODFLOW)
Finite difference grid

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
64	65	66	67	68	69	70	71	72

D. Reaches & Grav. Res. (GSFLOW)
Grid intersection of HRUs and segs.

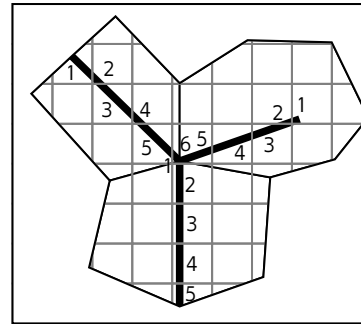



Figure 1. Major features of the GSFLOW geometry. **A.** Each segment is one link in the network. At each node, two tributary segments combine to flow into a single segment. Each is numbered. They need not be in any particular order, as indicated, but a downstream-increasing numbering scheme is required for updated inflows to all segments to be computed during the same iteration. **B.** Flow in each of the sub-basin HRUs is routed directly to a corresponding stream segment. The arrow on the upper left indicates that flow from outside of the representative tributary junction may also be part of the drainage network. Our topological approach to defining HRUs allows HRUs to be numbered the same as the stream segments that they enclose. Our code is written in such a way that future developments can relax this symmetry. **C.** MODFLOW operates on a grid that underlies the PRMS-based stream network and HRUs; each cell has a unique ID that is sequentially numbered. **D.** Gravity reservoirs are defined by the intersection of the PRMS HRUs and the MODFLOW grid. "Reaches" are defined as the section of each PRMS stream segment that lies within a single MODFLOW grid cell, and are numbered sequentially downstream as shown.

Specifically, within an HRU, the "soil zone" is subdivided into three reservoir types – the capillary reservoir, gravity reservoir, and preferential-flow reservoir, which are filled in order of increasing water storage using efficient water-accounting calculations (Section 2.1.2) (Figure 2). Underlying the PRMS soil zone are MODFLOW grid cells representing the deeper unsaturated zone and the saturated zone. While grid cells have uniform horizontal discretization, vertical layer thicknesses can be variable in order to accommodate different hydrostratigraphy. To link the PRMS and MODFLOW grids, the user must define gravity reservoirs at each different intersection of an HRU and a grid cell (Figure 1D). The MODFLOW component of GSFLOW also

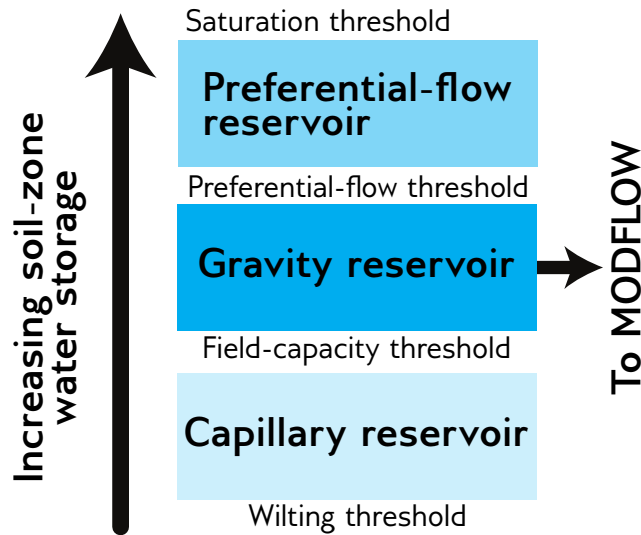



Figure 2. Soil-water storage reservoirs in the PRMS component of GSFLOW. Within each HRU, soil-water accounting calculations are carried out for three conceptual reservoirs in the order of increasing water storage and according to user-specified parameters. Climate forcing applies to the capillary reservoir, the gravity reservoir exchanges water with the deeper unsaturated and saturated zones represented by the MODFLOW component of GSFLOW, and Dunnian runoff and fast interflow occur in the preferential-flow reservoir. (Adapted from Markstrom et al., 2008, Figure 12.)

.

relies on a user-specified stream network; stream segments represent tributaries, and the intersection of a stream segment with MODFLOW grid cells defines stream reaches (Figure 1A, D).

GSFLOW uses a daily computational time step for both the PRMS component and MODFLOW component. Flows are exchanged between each component at each time step. Multiple MODFLOW "stress periods" can be invoked to represent different subsurface boundary conditions within a simulation period, but their lengths must be integer days.

2.1.2   Process description

This section includes a brief description of the main hydrologic processes represented in GSFLOW, with select parameters listed in Table 1. Full details can be found in the GSFLOW manual (Markstrom et al., 2008). In particular, Table 1 from Markstrom et al. (2008) summarizes all the surface-water processes captured by PRMS modules, groundwater processes captured by MODFLOW stress packages, and model coupling procedures captured by GSFLOW.

The PRMS component of GSFLOW includes modules that can convert commonly available climate data into complete forcing inputs needed for model simulations. These include methods for determining potential solar radiation, potential evapotranspiration, and snow accumulation or depletion; they also include different algorithms for spatially distributing data from one or a few observations points over the entire watershed.

Table 1. Select GSFLOW parameters (adapted from Markstrom et al., 2008, Appendix 1).

Parameter	Description
pref_flow_den	Decimal fraction of the soil zone available for preferential flow versus capillary zone flow
soil_moist_max	Maximum available capillary water-holding capacity of soil zone
soil_rechr_max	Maximum quantity of water in the capillary reservoir (value must be less than or equal to *soil_moist_max*)
soil_type	Soil type: 1=sand; 2=loam; 3=clay
soil_moist_max	Maximum volume of water per unit area in the capillary reservoir
slowcoef_lin	Linear flow-routing coefficient for slow interflow
slowcoef_sq	Non-linear flow-routing coefficient for slow interflow
ssr2gw_rate	Linear coefficient in the equation used to compute gravity drainage to MODFLOW finite-difference cell
ssr2gw_exp	Exponent in the equation used to compute gravity drainage to MODFLOW finite-difference cell
ssrmax_coef	Maximum amount of gravity drainage to MODFLOW finite-difference cell
sat_threshold	Maximum volume of water per unit area in the soil zone, between field capacity and saturation thresholds
hru_percent_imperv	Decimal fraction of HRU area that is impervious
ICALC	An integer value used to indicate method used to calculate stream depth in this segment
IRTFLG	An integer value that flags whether transient streamflow routing is active

For unsaturated zone flow, PRMS does not implement the Richards equation but instead applies computationally fast soil-water routing calculations to determine inputs and outputs for each HRU as well as exchanges among the three conceptual reservoir types within an HRU (GSFLOW manual Fig 19, Table 9). The "capillary zone" reservoir represents water held by capillary forces; it receives water through infiltration (based on parameter *pref_flow_den*) and loses water through evaporation

5 and transpiration (based on parameters *soil_moist_max*, *soil_rechr_max*, and *soil_type*). After reaching field capacity (parameter *soil_moist_max*), water transfers from the capillary zone to "gravity reservoirs", where water can flow horizontally as slow interflow (based on parameters *slowcoef_lin* and *slowcoef_sq*) or drain vertically into the deeper subsurface domain that is handled by MODFLOW (based on parameters *ssr2gw_rate*, *ssr2gw_exp*, and *ssrmax_coef*). Gravity reservoirs can also receive groundwater discharge from the MODFLOW component when hydraulic head values exceed the lower limit of the soil

10 zone. A fraction of gravity reservoir storage moves to the "preferential-flow reservoir" (based on parameters *pref_flow_den* and *sat_threshold*), where fast interflow occurs (based on parameters *fastcoef_lin* and *fastcoef_sq*). If the preferential-flow reservoir becomes full (based on parameter *sat_threshold*), then water exits the soil zone as Dunnian (saturation-excess) runoff. Horto-nian (infiltration-excess) runoff calculations apply for impervious fractions of HRUs (set by parameter *hru_percent_imperv*). Surface runoff and interflow are routed between HRUs, using a cascading flow scheme that follows user-specified indexing of

15 linked HRUs, and eventually reaches the stream network.

The MODFLOW component of GSFLOW computes water flow in the deeper unsaturated zone (UZF stress package), streams (SFR package), and saturated groundwater units (BCF, LPF, or UPW flow packages). Unsaturated zone flow is cal-culated using a kinematic-wave approach, which assumes that capillary (pressure gradient) flow is negligible compared to

gravity-driven flow. Capillary-dominated effects are instead represented in the soil zone of the PRMS component described above. Unsaturated zone flow in the MODFLOW component is calculated as waves representing wetting and drying fronts. Gravity reservoir drainage from the PRMS component flows to the top of the unsaturated zone of the MODFLOW component, unless the water table is above the soil-zone base – defined by the top of the MODFLOW domain – in which case the gravity reservoirs drain directly to the saturated zone. Saturated zone simulations (MODFLOW) employ the finite difference method to the groundwater flow equation.

Streamflow, as calculated by the MODFLOW component, includes inputs from upstream reaches, surface runoff and inter-flow from the PRMS component, base flow from the saturated zone discharge, and flows from possible underlying unsaturated areas. Outputs include flow to downstream reaches, leakage to groundwater, and flows to possible underlying unsaturated areas. Discharge across the streambed follows Darcy's law with specified streambed hydraulic properties. Five different options exist for stream discharge and head computations (parameter *ICALC*). The user can specify stream depths for each reach; apply Manning's equation to an assumed wide rectangular channel; apply Manning's equation for an eight-point-based channel and floodplain geometry; apply at-a-station power-law relationships between discharge, flow width, and flow depth (Leopold and Maddock, 1953); or specify an input look-up table of hydraulic geometries for each segment. Streamflow can be simulated as either steady-state flow (parameter *IRTFLG* = 0), where outflow to the next stream reach balances inputs, or as transient flow (parameter *IRTFLG* > 0), using a kinematic wave formulation for surface-water routing in channels, which applies the assumption that the water surface slope approximates the friction slope, and therefore negates backwater effects.

Some modifications were made to the original stand-alone PRMS and MODFLOW codes for their use in GSFLOW. Notably, the soil-zone structure of PRMS was significantly altered to facilitate its linkage with a MODFLOW subsurface domain. Other modifications are noted in the GSFLOW manual (Markstrom et al., 2008, see sections on "Changes to PRMS" and "Changes to MODFLOW-2005"). An additional feature starting in version 1.2.0 that is not described in the original manual is the inclusion of MODFLOW-NWT (Niswonger et al., 2011), a more numerically robust update to MODFLOW-2005 (Harbaugh, 2005) for groundwater flow.

2.2   GRASS GIS

GRASS GIS is an open-source, multi-purpose, and cross-platform geographic information system (Neteler and Mitasova, 2008; Neteler et al., 2008, 2012) that supports utilities for efficient raster and vector computations (Shapiro and Westervelt, 1994; Mitasova et al., 1995; Ŝúri and Hofierka, 2004; Hofierka et al., 2009). It includes both graphical and command-line interfaces, and may be driven by shell or Python scripts. It supports both 2D and 3D raster and vector data and includes SQL-based attribute table database management. GSFLOW–GRASS utilities are written for the most recent stable release version of GRASS GIS, v7.4. This supports Python scripting for both high-level built-in commands and for low-level access to database entries and vector geometries (Zambelli et al., 2013). We take advantage of these capabilities to develop an automated workflow to build GSFLOW inputs through GRASS GIS.

We chose GRASS GIS as the interface to develop inputs because (1) it is open-source and cross-platform; (2) it enforces rigid vector topology, which is critical for building stream networks; (3) its broad library of built-in hydrologic tools include

those for vectorized drainage network development with downstream-increasing indexing (Jasiewicz and Metz, 2011), which is essential for setting flow paths and adjacencies; (4) its generic Python scripting library and PyGRASS Application Programming Interface (API) make it easy to develop new extensions; (5) these extensions may be added to the official subversion (svn) repository, from which they can be automatically downloaded and installed on users' computers using the **g.extension** command; and (6) it provides a GUI and command-line interface (CLI) that are consistent with one another. The GUI and CLI interfaces are not required for GSFLOW–GRASS because the GRASS GIS component is handled mostly behind the scenes by a batch-processing Python script (**buildDomainGRASS.py**, Section 3.2); however, they allow end-users to re-run certain portions of the process and/or produce their own workflows using the GSFLOW–GRASS extensions as building blocks. The open-source aspect of the present work is in part motivated by the need for water assessment and planning tools in the developing world (Pal et al., 2007), and these extensions, combined with the interchangeable and consistent GUI and CLI, can help users to generate their own advanced customizations of GSFLOW–GRASS.

3   Methods

We adopt a heterogeneous surface and subsurface computational domain for GSFLOW–GRASS that employs sub-basin surface HRUs that are linked to subsurface grid cells. In addition to the computational efficiency of discretizing complex terrain into sub-basins with complex shapes rather than using a gridded surface domain at the resolution required to resolve the HRUs, the use of sub-basin HRUs that route surface runoff directly to stream segments also eliminates the need for establishing a cascading network (Section 2.1.2). Because of GSFLOW's conceptual (rather than gradient-based) surface-water-routing scheme, numerical differences between sub-basin and gridded HRU's are difficult to predict, but the automated GSFLOW–GRASS toolbox can help enable future testing to rigorously interrogate their respective performances.

GSFLOW–GRASS strikes a balance between generating a ready-to-go GSFLOW implementation and providing flexibility to customize applications. With a newly developed set of automated and robust GIS domain builder tools, GSFLOW–GRASS can be applied to any digital elevation model (DEM) to produce GSFLOW model simulations. Only a few steps are required to set up a GSFLOW model on the user's computer system. For further model-tuning, all scripts in the toolbox are open-source and commented to allow changes to any parameter as well as development of optional GSFLOW capabilities not included in the default GSFLOW–GRASS implementation. Many popular hydrologic model implementation programs have GUIs, including ModelMuse (Winston, 2009), Visual Modflow (Waterloo Hydrogeologic Inc., 2011), Hydrus (Simunek et al., 2009), ArcSWAT (Neitsch et al., 2002), and MIKE-SHE (Butts and Graham, 2005). While these are easiest for novice model users, GUIs can be challenging to develop for cross-platform implementations and generally support less flexibility for customization. Thus, we chose a mostly command-line approach, which has been designed and tested for use on Linux and Windows operating systems.

3.1   User-specified settings and model inputs

To seamlessly unify the different GSFLOW–GRASS functionalities, including the automated GRASS GIS domain builder, GSFLOW input-file builder, and visualization components, users specify model inputs and configurations using a Settings text

file. All inputs from the Settings file are read in and processed by the **ReadSettings.py** script. GSFLOW requires a daunting number of different model inputs (nearly 200 parameters for the PRMS sub-component alone). For ease of use, only a handful of application-specific and commonly adjusted inputs may be assigned using the Settings file, and default parameter values are applied elsewhere. While the default (and simplest) approach to GSFLOW–GRASS is to modify only the Settings file, other parameters (including those mentioned in Section 2.1.2) may be readily changed in the input-file builder by searching for the parameter names defined in the GSFLOW manual and changing their values. The open-source nature of our toolbox also allows users to add parameters to the Settings files for future extensions of GSFLOW–GRASS.

Specifying and including spatially variable properties is a major challenge to distributed modeling. The Settings file accommodates the use of variable aquifer hydraulic conductivity, channel width, and Manning's n parameters, which are described further in Section 3.3.3. Universal solutions are beyond the scope of the default toolbox, but we do provide a generalized GRASS-GIS extension called **v.gsflow.mapdata** to facilitate the generation of heterogeneous model inputs. **v.gsflow.mapdata**, further described in Section 3.2.4, can take any spatially variable data in a raster or vector GIS format and map it to one of the GSFLOW discretization structures: sub-basin HRUs for PRMS surface-water processes, regular grid cells for MODFLOW groundwater processes, gravity reservoirs that link the HRUs and MODFLOW grid cells, or stream segments or reaches for MODFLOW streamflow processes. This allows users to add data from any source – e.g., meteorological forcing, soil properties, hydrogeologic stratigraphy, or vegetation / land cover – to the GSFLOW–GRASS data structures. Other software tools have facilitated hydrologic modeling by automating the connection with established databases (Viger and Leavesley, 2007; Leonard and Duffy, 2013; Bhatt et al., 2014; Gardner et al., 2018). The USGS's GIS Weasel tool (Viger and Leavesley, 2007) may be used for deriving PRMS parameters from physical data sets such as STATSGO, which can then be mapped to the appropriate GSFLOW data structure using **v.gsflow.mapdata**. The current GSFLOW–GRASS release aims to provide a general set of tools and does not directly link with any specific databases, which are typically only available in observation-rich regions and countries.

The Settings file is divided into subsections, each of which drives a portion of the model setup and organization. The "paths" section defines the computer directory structure for the project and GSFLOW executable, as well as the project name and GSFLOW version. Three GRASS GIS sections, "GRASS_core", "GRASS_drainage", and "GRASS_hydraulics," set the GIS location and path to the DEM, the surface and subsurface flow discretization parameters, and open-channel flow geometry and resistance, respectively. The "run_mode" section allows the user to execute GSFLOW in either "spin-up" or "restart" mode (Regan et al., 2015). Spin-up simulations start with a preliminary MODFLOW steady-state execution using a specified infiltration rate (see below) to calculate reasonable initial groundwater head conditions for the subsequent transient simulation that includes both the surface and subsurface domains; the steady-state step can be essential for obtaining numerically convergent groundwater results and more realistic solutions for the entire coupled system. At the end of a spin-up run, final PRMS and MODFLOW state variables are saved in files that can be specified in the run_mode section to initiate "restart" coupled runs without the preliminary groundwater steady-state period. The "time" section is used to specify the temporal window of the simulation. The "climate inputs" section sets input parameters for the PRMS "climate_hru" option, which is the standard climate implementation supported by GSFLOW–GRASS (see Section 3.3.1) . Finally, the "hydrogeologic_inputs" section de-

fines the preliminary steady-state MODFLOW infiltration rate, used for "spin-up" runs, and either a layered or fully distributed subsurface hydraulic conductivity structure. The **ReadSettings.py** script uses these inputs to create a directory structure and organize all GIS and simulations files. This imposed directory structure supports easy exchange between the different toolkit modules and allows the use of relative directory names, which facilitates the sharing of model files across computers systems and between users.

3.2 GRASS GIS domain builder

A critical challenge for any distributed hydrologic model is the fully automated development of a reproducible, topologically correct, and interlinked data structure that describes water flow through a catchment in a computationally efficient manner. Semi-automated approaches to building surface flow networks are common (e.g., Luzio et al., 2006; Arnold et al., 2012), but the development of a fully automated approach has been impeded by the mathematical and logistical difficulties of building a topologically ideal drainage network (i.e. one whose fundamental unit is a tributary junction). Many standard GIS tools encounter problems when handling complex digital topography (represented using a DEM) that may contain natural or artificial depressions and whose grid cells are often much larger than real topographic features. Further complications arise when incorporating surface flow networks into integrated hydrologic models, because each link within the network must then be tagged with sufficient information to identify drainage pathways through the whole network, and the stream network must also be linked with sometimes different geometries and resolutions for surface-water and the groundwater-flow grids.

We addressed this challenge by creating eleven new GRASS GIS "extensions," also called "add-ons," that work alongside core GRASS GIS commands to transform user inputs (including a single DEM) into a set of GSFLOW inputs. This workflow creates an automatically generated network of streams and HRUs that is spatially linked to a MODFLOW grid. The domain-building procedure is automated through the **buildDomainGRASS.py** script, which takes inputs from the Settings text file, implements the domain-building workflow, and produces ASCII files used by GSFLOW–GRASS's Python input-builder scripts.

3.2.1 Surface-water network

In the first step of the fully automated domain-building workflow, GRASS GIS imports a user-provided DEM to define the drainage network and HRUs. After hydrologically correcting the DEM by filling pits and removing cells that have flow inputs from outside the map area (GSFLOW–GRASS requires the full topographical catchment to be included in the model domain), a Hortonian drainage network is constructed using the **r/v.stream.*** toolkit (Jasiewicz and Metz, 2011) that relies on a single-flow-direction implementation of the **r.watershed** flow-routing algorithm (Metz et al., 2011). Sub-basins associated with each stream segment are designated as HRUs in order to follow both the natural discretization of the landscape and the architecture of PRMS (Markstrom et al., 2015). River headwaters are defined based on a threshold drainage area that may be weighted by the user to represent, for example, nonuniform precipitation or snowmelt inputs. Such weights permit a more realistic representation of drainage density and, as a result, increased model resolution in areas that contribute more water to the catchment.

The GRASS GIS drainage-network-creation algorithm, **r.watershed** (Metz et al., 2011), is both efficient and accurate. For computations that can take place entirely within memory, its speed exceeds that of both Terraflow and the D8 routing used by ArcGIS (Jenson and Domingue, 1988; Maidment and Morehouse, 2002; Arge et al., 2003; Magalhães et al., 2014). This speed results from its sorting algorithm and priority queue, and a standard desktop workstation today can process DEMs in memory with tens of thousands of cells on each side. The least-cost path approach taken by **r.watershed** does not require any pit-filling step, but we do so in order to create a more consistent DEM with downslope-routed flow for the remainder of the analysis. The flow-routing component of the more recent "Fastscape" algorithm by Braun and Willett (2013) could be faster than **r.watershed**, but these have not been benchmarked, and Fastscape is not yet integrated into the GRASS GIS toolchain, which is necessary for all of the subsequent steps. Kinner et al. (2005) demonstrated that **r.watershed** is more accurate than Terraflow (Arge et al., 2003), especially in low-relief areas and those in which tree canopy elevations are mistakenly interpreted as ground-surface elevations; this latter issue is pervasive across many digital elevation models, including the widely used Shuttle Radar Topography Mission (SRTM) DEM (Farr et al., 2007; Miliaresis and Delikaraoglou, 2009).

In spite of these advantages, **r.watershed** has not before been used to build flow networks for integrated hydrologic models. Other integrated hydrologic model domain-building tools use local drainage direction information (Bhatt et al., 2014; Maxwell et al., 2017; Gardner et al., 2018). While not an integrated hydrologic model due to its limited subsurface modeling capabilities, Srinivasan and Arnold (1994) integrated GRASS GIS version 4, including an earlier and much slower version of **r.watershed**, into the Soil & Water Assessment Tool (SWAT). Beyond this, **r.watershed** is typically discussed in the drainage algorithm literature (e.g., Barnes et al., 2014; Magalhães et al., 2014; Schwanghart and Scherler, 2014; Sangireddy et al., 2016), directly applied to flow-routing and cost-path calculations (e.g., Wickert et al., 2013; Bird et al., 2016), or included as a component of an assessment tool (e.g., Bhowmik et al., 2015; Rossi and Reichenbach, 2016). By integrating **r.watershed** into GSFLOW–GRASS, via the **r/v.stream.*** toolkit for Hortonion drainage network analysis (Jasiewicz and Metz, 2011), we are able to harness the capabilities and efficiency of the hydrologic computational engine within GRASS GIS for integrated hydrological modeling.

Following drainage network construction, the next step in the automated workflow is to map the connections between each segment in the tributary network. To do this, we developed an extension called **v.stream.network**, which builds atop the upstream-to-downstream stream-segment and HRU indexing in the existing **r/v.stream.*** toolkit (Jasiewicz and Metz, 2011). This index is a unique positive integer identifier applied to each segment and its overlapping HRU, and is called a "category" in GRASS GIS. For each segment and overlapping HRU in the drainage network (Figure 1A,B), **v.stream.network** writes the category value of the immediately downstream stream segment to the "tostream" column in its associated attribute table row. Any stream segment exiting the map area is given a "tostream" value of 0. This links the stream segments and HRUs in the watershed as a directed graph (e.g., Czuba and Foufoula-Georgiou, 2014; Heckmann et al., 2014; Tejedor et al., 2015).

At this point, the user may optionally break out of the automated workflow and edit the vector geometries that define the streams and sub-basins. While we expect that many users will find the fully automated approach to be a major advantage over those that require manual intervention – these add a source of subjectivity and laborious processing time – Gardner et al. (2018) note that human-developed drainage structures may cause a discrepancy between topographically routed flow and actual flow

paths. This manual adjustment is not standard, and requires the addition of a break point in **buildDomainGRASS.py**, as well as for the user to manually adjust the category numbers (indices) and "tostream" network topology values in the attribute tables for the segments and HRUs if the changes are substantial enough to change the flow network.

After this, the study area is limited to a single drainage basin using the new **v.stream.inbasin** extension, completing the development of the drainage network geometry and topology. This step is included because the goal of many hydrologic studies is to understand a single watershed basin. If this is not the case, **buildDomainGRASS.py** may be edited to skip this step and to analyze all complete drainage networks within the domain.

Each stream segment is then supplied with attribute values required for GSFLOW through the **v.gsflow.segments** extension. This numbers each segment for GSFLOW (Figure 1A) and populates the associated database table with hydraulic geometry, channel roughness (constant or spatially distributed), and channel and floodplain width (constant or spatially distributed). Additional less-commonly used options are also available, including additional input discharge for the upstream-most stream segments (e.g., from human intervention), input diffuse runoff, and direct precipitation on the stream.

3.2.2 Groundwater-flow grid

Following the completion of the surface-water domain, the next step is to build the groundwater domain. MODFLOW-NWT uses a rectangular finite-difference grid structure (Harbaugh, 2005; Niswonger et al., 2011). The cell size for this grid is selected by the user in the Settings file. It is often necessary to discretize the MODFLOW groundwater domain on a grid that is coarser than the DEM used for surface flow routing in order to increase computational efficiency while still allowing GSFLOW–GRASS to generate a complex surface-water network; the proper grid cell size depends on the size of the HRUs and the strength of the surface-water–groundwater coupling. **v.gsflow.grid** builds the MODFLOW grid as a set of GIS vector areas (Figure 1C) using the built-in **v.mkgrid** command. The resolution of this grid is approximately that desired by the user and meets two criteria designed to most accurately and efficiently capture the topographic information represented in the fine-scale flow-routing DEM. First, the grid size must allow for MODFLOW grid-cell edges to align perfectly with flow-routing DEM cell edges. Second, the grid size must be able to generate a domain that fits exactly within a tight-fitting bounding box snapped to the closest flow-routing DEM cell edges outside of the study watershed. As a final step, this tight-fitted domain is padded by three cells on each side to ensure appropriate boundary-condition handling.

r.gsflow.hydrodem then performs a hydrologically correct resampling of the original flow-routing DEM to the resolution of the MODFLOW grid cells. This resampling is required when users desire a MODFLOW grid that is coarser than the flow-routing DEM for computational feasibility. Without a hydrologically-correct resampling, MODFLOW cells would be assigned the overall mean elevation from the corresponding region of the flow-routing DEM. In this case, MODFLOW grid cell elevations may average across valley floors and valley walls, creating a bumpy river longitudinal profile that contains artificial dams (Figure 3A). With the hydrologic correction, MODFLOW cells that do not contain stream segments remain unchanged, but cells containing stream segments are assigned the mean elevation of only the river-channel cells in the flow-routing DEM. This enforces decreasing elevations down the drainage network, and Figure 3B demonstrates the resulting continuous flow through the catchment.

om

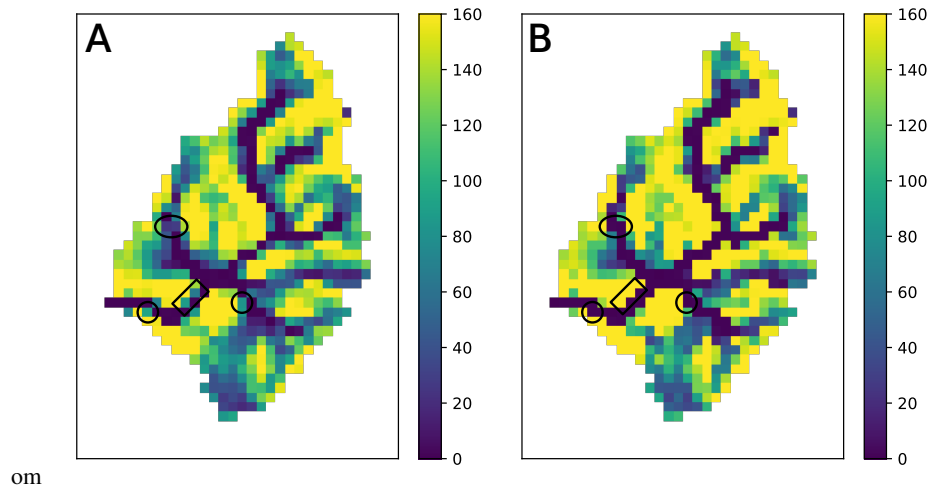



Figure 3. Water table depths (A) without and (B) with hydrologic correction. Circled areas display where the hydrologic correction produced continuous shallow groundwater through the the channel network. Rectangles indicate where the hydrologic correction allowed water held behind artificial dams to drain. This example is from the Shullcas River watershed (Section 4.3).

.

3.2.3 Surface-water–groundwater coupling

The final step in developing the GSFLOW domain is to link the surface-water geospatial data structures (HRUs and segments) with the MODFLOW rectangular grid. **v.gsflow.reaches** and **v.gsflow.gravres** construct the reaches and gravity reservoirs (Section 3.1), which are the intersection of segments and HRUs, respectively, with each MODFLOW grid cell (Figure 1D). The database table for the reaches includes values for the thickness of the stream-bed sediment (defaults to 1 m) and its hydraulic conductivity (defaults to 5 m/d, characteristic of sand and gravel).

3.2.4 Accessing additional GSFLOW functionality

GSFLOW supports more input options than we have defined for our GRASS GIS **v.gsflow.*** commands, though we have included many of the most common options. These are sufficient to set up and run a GSFLOW simulation. However, they may not encompass all of the variables that some users may consider to be important.

Therefore, GSFLOW–GRASS includes the **v.gsflow.mapdata** tool for users to add other attributes to database tables, with a focus on spatial distributions. These attributes can include spatially variable precipitation and temperature, parameter choices for model spin-up, and fully distributed maps of hydraulic conductivity, specific yield, streambed hydraulic properties, soil texture, vegetation type, and evapotranspiration parameters. The core capability of **v.gsflow.mapdata** is the use of averaging and nearest-neighbor methods to connect input data from raster grids, vector areas (polygons), or vector points, to the attribute tables of the HRUs, segments, gravity reservoirs, reaches, and/or MODFLOW grid cells. As these are custom additions, calls to

v.gsflow.mapdata must be added by end users to **buildDomainGRASS.py**. Once added, the end user can follow our template code in the input-file builder to add these to the GSFLOW input files. **v.gsflow.mapdata** therefore adds user-driven flexibility in which input data can be supported by GSFLOW–GRASS, and a starting point for users who may want to expand on its capabilities.

3.2.5   Geospatial data export

In the final step, the generated attributes and geometries are exported. This information is stored in GRASS GIS as raster grids and vector geometries associated with SQL database tables. **buildDomainGRASS.py** exports a rasterized "basin mask" (1 in the basin, 0 outside) and the hydrologically corrected DEM at the MODFLOW grid resolution, as well as vectorized GIS data (shapefile format) for the HRUs, gravity reservoirs, MODFLOW grid, stream segments, stream reaches, pour point, full study basin area, and downstream boundary-condition cells. **v.gsflow.output** exports the database tables associated with the vectorized GIS data in comma-separated variables (CSV) files that can be read in by the input-file builder scripts (Section 3.3) for use in GSFLOW. These exported data are then ready to be parsed into GSFLOW inputs using the Python input-file builder scripts (Section 3.3) and/or to be used for data visualization (Section 3.5).

This separation between the GIS and ASCII-input-file components is intentional. The GRASS domain-builder component typically requires several minutes to run, and often only needs to be executed once for a watershed. The ASCII files, on the other hand, can form an effective basis for ensembles of runs. These can be used to calibrate parameters or explore hydrologic sensitivity to variable forcing scenarios.

3.3   GSFLOW Input File Builder

GSFLOW–GRASS includes a set of input-file builder scripts that are streamlined to incorporate the model domain discretization constructed by the GRASS GIS workflow and generate corresponding model inputs for the GSFLOW control file, PRMS-type input files, and MODFLOW-type input files. Most of the new features in GSFLOW that are not in stand-alone PRMS or MODFLOW follow the same Modular Modeling System input-data file format (Leavesley et al., 1996) as PRMS, which includes use of a "control file" as the main interface file, "modules" for different computational options, and the PRMS input file syntax. In contrast, MODFLOW uses a "name file" as its main interface file, implements "packages" for computational options, and follows its own file syntax. The following builder scripts handle these different formats and are automatically executed through the toolkit's Run file (Section 3.4). The builder scripts may also be customized for extensions beyond the default implementation.

3.3.1   GSFLOW control file

The GSFLOW control file is the highest level input file and is created by the **printGSFLOWControlfile.py** script in the GSFLOW–GRASS toolkit. The toolkit is streamlined for configuring the integrated mode of GSFLOW (set through the "model_mode" parameter).

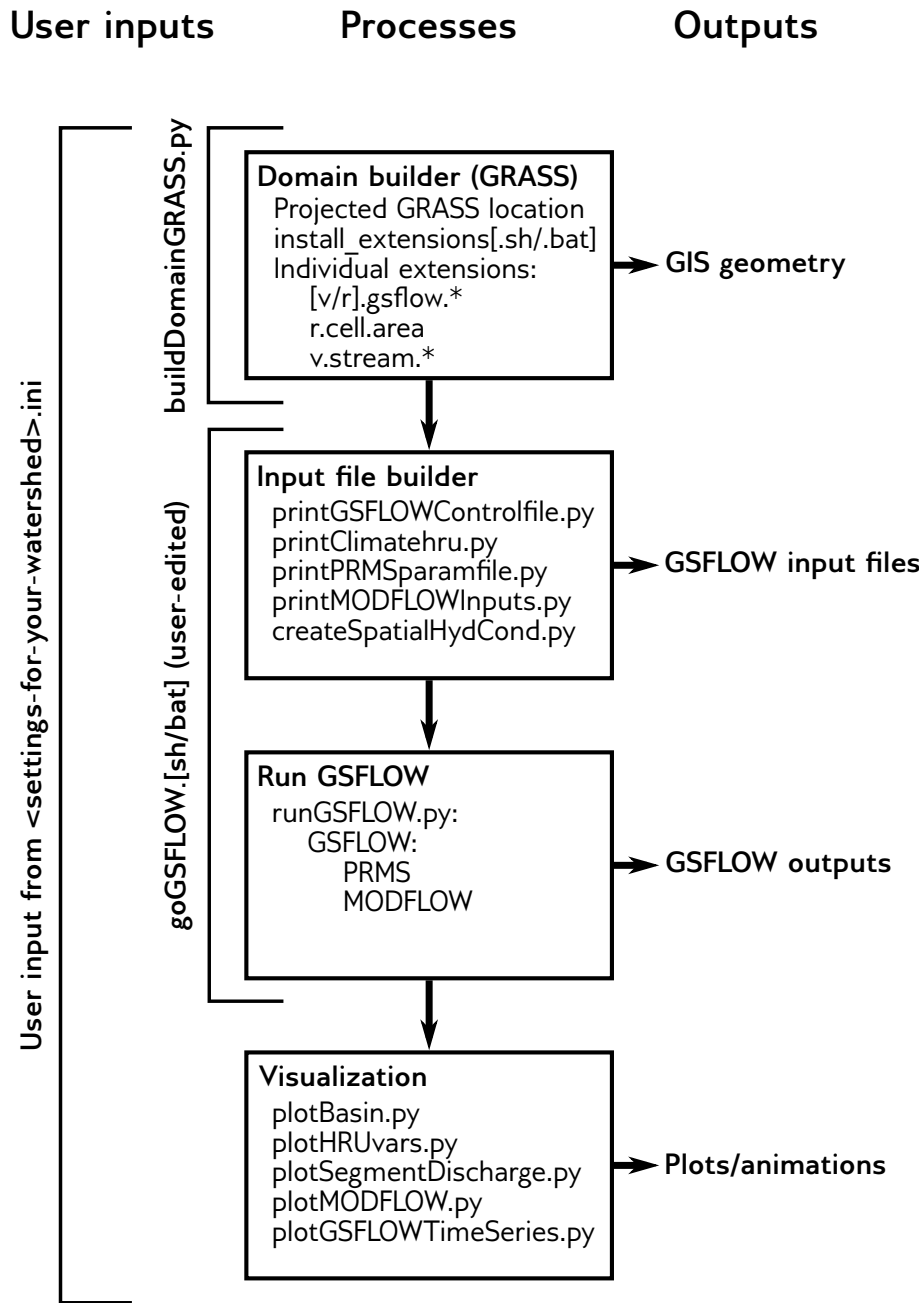



Figure 4. GSFLOW–GRASS workflow. The user: (1) creates a ***.ini** file based on their study catchment; (2) creates a projected GRASS GIS location; (3) runs **buildDomainGRASS.py**; (4) edits and runs **goGSFLOW.py**. After this, they may use GSFLOW–GRASS's visualization tools to study the GIS and model outputs.

.

Inputs for the control file parameters are organized under six numbered sections in **printGSFLOWControlfile.py**. The script sets parameters related to climate forcing, time domain, and run mode based on what the user specifies in the Settings file; all other parameters are pre-set to default values. Further customization of control file parameters (stored in the list variable *con_par_name*) requires simply changing default values (in the corresponding list variable *con_par_values*) in the script;

5   spatially variable entries can be generated with the aid of the **v.gsflow.mapdata** tool. The first two sections are required and include details about the simulation execution and module choices. The third section establishes spin-up versus restart run modes based on Settings file entires. Sections 4 and 5 contain customizable lists of output variables to be printed, which can be used by visualization scripts in GSFLOW–GRASS (Section 3.5). The last optional section is for running the model in a debugging mode.

10 Note that the default implementation of this toolkit uses the "climate_hru" module for precipitation and minimum and maximum daily temperature; this means that the model will employ pre-existing files containing data already specified by HRU. The PRMS component of GSFLOW does include other modules for distributing data from one or a handful of weather stations, but these typically require application-specific empirical parameters that are difficult to incorporate in a generic toolkit. Use of the "climate_hru" module provides flexibility for the user to implement their own spatial interpolation or extrapolation methods,

15 which can then be transferred to the GSFLOW domain with the **v.gsflow.mapdata** tool. GSFLOW–GRASS's default implementation also uses the Priestley-Taylor formulation for potential evapotranspiration calculations (Markstrom et al., 2008). This module was chosen because of its reliance on only air temperature and solar radiation (calculated by the PRMS component of GSFLOW), and because of the relative ease of accounting for different vegetation properties through the parameter *pt_alpha* (in the PRMS parameter file, Section 3.3.2).

20 After the six parameter input sections in **printGSFLOWControlfile.py**, the script builds the control file and then generates an executable file (shell script for Linux or batch file for Windows) for running GSFLOW with the control file. After all other input files are created, this executable is called by the toolkit's automated Run file (Section 3.4). The executable can also be used to run GSFLOW outside of the GSFLOW–GRASS toolkit.

3.3.2   PRMS-type input files

25 Input files required for the PRMS component of GSFLOW are the parameter file ("param_file" in the control file), which includes empirical surface and soil zone properties, and the data file ("data_file" in the control file), which includes climate observations for the spatial interpolation/extrapolation algorithms. If the "climate_hru" module is selected, as it is in the toolkit's default implementation (Section 3.3.1), then individual input files with HRU-distributed climate variables must also be specified. For a quick set-up of GSFLOW–GRASS, the script **printClimatehru.py** takes daily observations from a single file

30 and distributes them uniformly over all HRUs. The toolkit handles the minimum required climate variables – daily precipitation, maximum temperature, and minimum temperature, and it is set up to be readily extended to also include humidity, solar radiation, and/or wind speed. A spatially uniform approach may be acceptable where the size of a rainstorm is typically greater than the size of a catchment and climatic variables vary only weakly with slope and aspect. Larger and higher-relief catchments

require spatially distributed climate inputs for realistic outputs; these require custom inputs from the end-user, which can be ported from any discretization to the HRU domain with the aid of the **v.gsflow.mapdata** tool.

The parameter file is created by the script **printPRMSparamfile.py**. The script includes sections for domain dimensions and for parameters inputs, both of which are streamlined to take values parsed from the GRASS GIS domain builder outputs
5 (as indicated in the comments in the script). Because of PRMS's conceptual soil moisture regimes, the parameter file requires a substantial number of parameter inputs related to the soil and vegetation that cannot easily be specified without calibration. As a default to help the user get GSFLOW up and running, most parameter values in **printPRMSparamfile.py** are preset, mostly using calibrated values from the Sagehen watershed example that was distributed with the GSFLOW model version 1.2.1. We have indicated with the comment "# *** CHANGE FOR SPECIFIC SITE" those parameters that could also be altered based on
10 known characteristics of one's watershed site. This includes various soil and land-cover inputs, such as *soil_type* (sand, loam, or clay), *cov_type* (bare soil, grasses, shrubs, or trees), *transp_end* (end month of transpiration, for phenology), and *pt_alpha* (Priestley-Taylor parameter α, which can be based on literature values). In addition to these highlighted parameters, users can review all parameters to determine whether others could be particularly important for their specific application. These may include some of the parameters mentioned in Section 2.1.2 that determine exchanges between different soil-zone reservoirs.
15 Spatially variable information can be transferred to the HRU domain using the **v.gsflow.mapdata** tool. Rigorous calibration of PRMS parameters can eventually be carried out with inverse codes such as PEST (Doherty, 1994) or UCODE (Poeter and Hill, 1998, 1999).

3.3.3 MODFLOW-type input files

GSFLOW requires input files for each MODFLOW package utilized, which can include any of the packages listed in Table 1
20 of the GSFLOW manual (Markstrom et al., 2008, Appendix 1, p. 176-226 provides details). Our toolkit by default creates a relatively general MODFLOW set-up, which includes required input files and omits most optional ones, such as the Well package. Our Python library **MODFLOWLib.py** consists of functions for creating: four Basic package input files (name file, basic package file, discretization file, and the optional output control file for customizing output files), two different groundwater flow package options (the Layer-Property Flow (LPF) from MODFLOW-2005 and the Upstream Weighting Package (UPW)
25 from MODFLOW-NWT), the numerical solver package (Preconditioned Conjugate Gradient (PCG) for LPF and Newtonian (NWT) input file for UPW), the Streamflow-Routing package (SFR), and the Unsaturated-Zone Flow package (UZF).

The script **printMODFLOWInputs.py** calls the functions from **MODFLOWLib.py** to create a set of internally consistent input files that incorporate the domains constructed by the GRASS-GIS workflow (Section 3.2) and conform to the simulation directory structure established through the **ReadSettings.py** utility. By default, **printMODFLOWInputs.py** calls the
30 MODFLOW-NWT UPW/NWT flow package instead of the MODFLOW-2005, because of the superior numerical performance of the former in tests with steep elevation gradients (e.g., Section 4.3). If desired, users can easily switch to the LPF/PCG formulation from MODFLOW-2005 by setting *sw_2005_NWT* = 1 in **printMODFLOWInputs.py**.

Input files created outside of our toolkit for a stand-alone MODFLOW model implementation of identical discretization will for the most part be usable with the integrated GSFLOW model. However, as indicated in Table 1 of the GSFLOW manual,

some MODFLOW packages were modified for their use in GSFLOW. Advantages of implementing our toolkit over using pre-created MODFLOW input files are that it already incorporates these GSFLOW modifications, it automatically uses the GRASS-GIS builder results for the domain, and it guarantees a directory structure that is consistent with the rest of the input files and the visualization scripts.

5   The GSFLOW–GRASS toolkit also offers an optional script **createSpatialHydCond.py** for generating spatially distributed hydraulic conductivity fields for the upper layer based on elevation and/or distance from the stream network, with the assumption that lower elevations and/or riparian corridors have higher hydraulic conductivity properties. Because application-specific entries cannot easily be generalized for input through the Settings file, users should directly customize elevation and stream distance thresholds, as well as corresponding hydraulic conductivity values, at the top of the **createSpatialHydCond.py** script.

10 This script will automatically import domain information from the Settings file and export results to the file location specified by the Settings file. **createSpatialHydCond.py** serves as a ready-to-go tool for creating physically plausible hydraulic conductivity patterns, and it provides an example for how users can create their own scripts to customize spatially distributed inputs. A similar type of script could create spatially distributed infiltration fields for the preliminary MODFLOW steady-state simulation in spin-up runs (e.g., *finf* entry in the Settings file). These tools can provide preliminary inputs to jump-start GSFLOW

15 model implementations. However, realistic construction of hydrogeologic frameworks relies on data from sources such as well logs, geologic maps, geophysical measurements, and pumping tests (Reilly, 2001; Reilly and Harbaugh, 2004). For these, we recommend that users import the appropriate data sources into GRASS GIS and use the **v.gsflow.mapdata** extension to map these parameters onto the appropriate GSFLOW objects (e.g., HRUs, MODFLOW cells). Properties for stream segments and reaches – such as streambed hydraulic conductivity, and unsaturated hydraulic properties below the streambed – are set to

20 default values that can be changed through the GRASS GIS extensions. By default, the streamflow calculation is set to use Manning's equation by assuming a wide rectangular channel (*ICALC*= 1). Spatially variable stream widths and/or Manning's n values may be set through the Settings file, based on either gridded or point-based (e.g., survey) data, and **v.gsflow.segments** also supports the delineation of both channel and floodplain geometries and roughness parameters.

3.4   GSFLOW run file

25 For the user's convenience, the GSFLOW–GRASS toolkit includes an executable Run file, which is a shell script for Linux, **goGSFLOW.sh**, and a batch file for Windows, **goGSFLOW.bat**. The Run file collects input from a specified Settings file and then runs all of the above input-file builder scripts; the script **runGSFLOW.py**, which launches the GSFLOW simulation; and the runtime visualization script **plotGSFLOWTimeSeries_Runtime.py**, further described below. If the runtime visualization is not desired, the user can comment out the corresponding execution line in the Run file. As long as the user does not wish to

30 use more features than are exposed in the Settings file, no direct interface with the code is required to run GSFLOW–GRASS. This permits a "quick-start" implementation of GSFLOW, which can substantially lower the barrier to entry for using this model.

The Run file may be implemented only after the model domain is generated through **buildDomainGRASS.py**. The GSFLOW–GRASS toolkit separates the GRASS domain-builder module from the Run file because users will typically only need to con-

struct their domain once, but will perform multiple runs of the model with variable parameter inputs, for example, for model calibration or to simulate different time periods.

After preliminary quick-start simulation tests, users can further customize their runs by taking advantage of the modular structure of the toolkit, which has a separate script for each input file. For example, to target specific aspects of the model, such as the surface runoff properties, corresponding parameters may be adjusted in the PRMS parameter file by editing and re-running **printPRMSparamfile.py**. Select input-file builder scripts can be run either within Python, or by editing the executable Run file.

3.5 Visualization tools

Our toolkit includes post-processing Python scripts that employ the Matplotlib plotting library (Hunter, 2007) for visualizing the domain discretization, key MODFLOW inputs, and model output results. The model discretization for the PRMS component of GSFLOW is exported from GRASS GIS as a set of standard vector GIS files (shapefiles). Our Python plotting scripts use these shapefiles to create figures of the surface HRU and stream segment discretization (**plotBasin.py**), and to generate movies of HRU-distributed and stream segment-distributed variables (**plotHRUvars.py** and **plotSegmentDischarge.py**). These output variables (e.g., evapotranpsiration and streamflow) are set through *aniOutVar_names* in the GSFLOW control file (see Section 3.3.1). The exported shapefiles may also be used to visualize model results with standard GIS packages (e.g., QGIS: QGIS Development Team, 2013) outside of GSFLOW–GRASS.

For the MODFLOW component of GSFLOW, the toolkit's script **plotMODFLOW.py** plots spatially distributed layer elevations, hydraulic conductivity, and a map of active computational grid cells. The script also plots spatially distributed MODFLOW simulations results over time, including for hydraulic head, change in head, water table depth, and recharge from the unsaturated zone. For storage efficiency, the toolkit creates and reads in head and unsaturated zone output files in binary format.

For basin-total GSFLOW results, the Python script **plotGSFLOWTimeSeries.py** generates time series lines for user-selected variables from the main GSFLOW CSV output file. Names of all variables, along with their descriptions and units, are listed in **GSFLOWcsvTable.py**, which is imported into **plotGSFLOWTimeSeries.py** to ensure consistency in figure labels and axes. Our toolbox also includes the runtime visualization script **plotGSFLOWTimeSeries_Runtime.py** that is by default called by the Run file (but can be commented out if desired) and displays a continuously updated time series plot of basin-total precipitation and discharge. Tracking simulation progress with runtime plots can be very useful for complex integrated models, which can have lengthy simulation times.

The visualization scripts can be run using a command-line parser and/or by editing plot options that appear near the top of each script. More advanced users may modify the bodies of the scripts to change to features such as axis intervals or color schemes. For users who want to adjust the scripts, we suggest running them in the iPython interactive programming console (Pérez and Granger, 2007), which is also incorporated into the Spyder integrated development environment (IDE). Although this visualization approach requires some familiarity with Python and/or command-line argument parsing, it accommodates a wide range of plotting preferences. All plots and videos may be displayed as on-screen figures (in raster or vector formats,

Figure 5. Our test sites include the high Andes, a mountainous island, and a formerly glaciated Mississippi tributary.

using the interactive Matplotlib window), and may be saved as images (interactively) or videos (***.mp4** format) as defined by inputs to the plotting script.

Other existing no-fee USGS GUI programs for MODFLOW also provide visualization capabilities, and using these with the input and output files produced with GSFLOW–GRASS is possible. In particular, GW Chart (Winston, 2000) can be
5 directly implemented for plotting basin-level time series results. Additionally, Model Viewer (Hsieh and Winston, 2002) and ModelMuse (Winston, 2009) are able to read in and plot spatially variable head results from binary files with the extension ".bhd," but this does require manual post-processing steps. For Model Viewer, the user needs to copy all MODFLOW input and output files to a new folder inside the Model Viewer project directory and select the namefile when prompted. For Model Muse, the user must first delete the line that starts with "IWRT" from the name file in order to load the project into the program.
10 Once the project settings are loaded into ModelMuse, the user can use the "import model results" tool to select the binary head file.

4   Examples

Three example implementations demonstrate (1) the variety of hydrological processes and environments that can be explored using GSFLOW–GRASS, and (2) how the toolkit's GIS domain builder can handle diverse topographic settings, including those
15 prone to problems with standard GIS stream network tools. Towards the first point, the specific examples chosen represent a range of practical applications for water and land management. Towards the second point, each simulation presents a unique

Table 2. Catchment and hydrological characteristics of the GSFLOW–GRASS example sites.

Site	Drainage area [km^2]	Flow-routing cellsize [m^2]	MODFLOW cellsize [km^2]	Min. HRU area [km^2]	Elevation range [m]	Mean annual rainfall [mm]	Daily rainfall CV
Cannon River, Minnesota, USA	3723	225	1	10	203–413	756	3.2
Water Canyon, Santa Rosa Island, California, USA	12.5	8100	0.0324	0.4	23–378	265	5.4
Shullcas River, Junín Region, Peru	161.4	930.93	0.25	1	3526–5527	1076	1.4

Precipitation statistics from 1938-05-12 to 1943-11-05 (Cannon); 1990-04-23 to 2017-09-27 (Water Canyon); 2013-08-26 to 2016-09-29 (Shullcas). "Flow-routing cellsize" is the original DEM resolution used to construct the stream network and irregular HRU cells, which are ultimately coarser-sized ("Min. HRU area"). CV = coefficient of variation.

set of technical challenges in developing a topographically based model domain that can properly route rainfall through a network of stream segments and sub-basins as well as a connected groundwater-flow grid. It is important to note, however, that no calibration effort was made to match field observations for these test cases. The simulation results thus serve as purely schematic examples based on certain settings and do not aim to capture actual conditions at the corresponding sites. The three examples all contain complex hydrology with interactions between surface water and groundwater and are exemplars of practical management concerns. Together they span a range of environments: high to low elevations, steep to low-gradient catchments, coastal to inland settings, tectonically active to cratonal, and with partially to fully integrated drainage. Their catchment areas range from 12.5 km^2 to 3723.0 km^2, covering the range of scales that GSFLOW was developed to simulate.

Figures 6, 8, and 9 display sample inputs and outputs of the model simulations using the default GSFLOW–GRASS toolkit for the three test cases. These applications show that even before any parameter adjustments, the GSFLOW–GRASS toolkit can readily generate GSFLOW model domains and parameter inputs that produce numerically convergent simulations in a variety of topographies and hydroclimatic conditions.

Preliminary simulations with the default GSFLOW–GRASS provide a valuable springboard for the next step of performing the calibration needed to generate realistic model outputs for specific sites. The GSFLOW–GRASS toolbox can be customized to quickly generate additional model runs with varying input values to expedite the parameter calibration. It can also facilitate the implementation of sensitivity or other Monte Carlo-type analyses that are critical for identifying issues such as equifinality and over-parameterization and for determining uncertainty estimates (Beven, 2006; Gallagher and Doherty, 2007; Razavi and Gupta, 2015; Song et al., 2015).

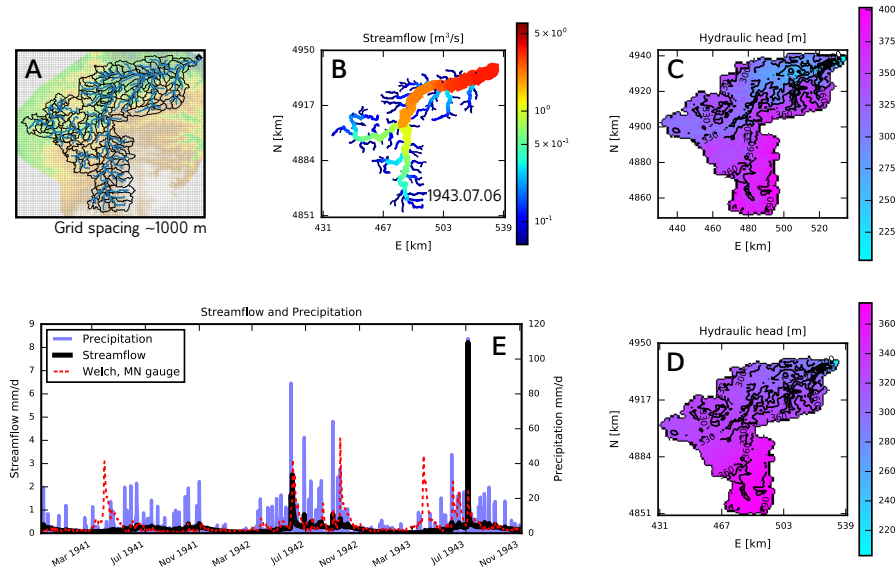



Figure 6. Model based on Cannon River, Minnesota, USA. **(A)** Map with MODFLOW grid, HRU outlines, stream segments (blue), and digital elevation model. **(B)** Simulated discharge after an 11 cm rainfall event. **(C, D)** Relatively low-gradient hydraulic head distributions in two MODFLOW layers representing an upper glacial till unit (low hydraulic conductivity) and lower fractured carbonate bedrock (higher hydraulic conductivity), with elevation contours (m.a.s.l.). **(E)** Three-year hydrograph showing uncalibrated discharge simulations matching observations reasonably well during non-peak flood times but poorly during many of the actual peaks.

4.1 Cannon River, Minnesota, USA

The Cannon River is a tributary to the upper Mississippi River in Minnesota, USA. Its headwaters cross low-relief uplands that are capped by low-hydraulic-conductivity glacial deposits (Patterson and Hobbs, 1995) and are intensively farmed (Kreiling and Houser, 2016). Its lower reaches pass through a valley cut into fractured carbonate bedrock that is popular for recreation.

5   This combination of agricultural and recreational uses leads to a suite of management concerns related to agricultural nutrients and fine sediments (Czuba and Foufoula-Georgiou, 2014) impacting both surface water and the underlying bedrock aquifers (Tipping, 2006; Steenberg et al., 2013), thus motivating the need for integrated hydrologic modeling tools. Furthermore, its large basin and high-resolution topography make the Cannon River drainage basin an appropriate test site to evaluate the accuracy and computational efficiency of GSFLOW-GRASS across a range of HRU and MODFLOW grid cell areas.

10   We implemented GSFLOW–GRASS for the Cannon River watershed using the Minnesota statewide 1 m LiDAR data set (http://www.mngeo.state.mn.us/chouse/elevation/lidar.html), which we resampled to 15 m resolution. Meteorological data from nearby Zumbrota, Minnesota was obtained from the Midwestern Regional Climate Center (Adresen et al., 2014). The flexible GSFLOW–GRASS input builder allows for easy implementation of two MODFLOW layers to represent an upper

glacial till unit (low hydraulic conductivity) and the underlying fractured carbonate bedrock (higher hydraulic conductivity), which produced somewhat steeper hydraulic head gradients in the upper layer compared to the lower layer (Figure 6C-D).

In this low-relief watershed (Table 2), Pleistocene glaciation produced non-integrated upland drainage that presents computational challenges that are very different from those in steep watersheds. Much of the Cannon River watershed's post-glacial topography is characterized by small localized hills and enclosed basins that have not yet been organized (i.e., integrated) by fluvial erosion into a linked valley network, in which water flows directly to a stream without encountering an enclosed depression (such as a lake, wetland, or dry basin). In such settings that lack integrated drainage, downslope flow-routing and "pit-filling" algorithms that are typically used to build hydrologic model domains (e.g., Bhatt et al., 2014; Maxwell et al., 2017; Gardner et al., 2018) can fail or produce spurious results by inappropriately modifying the real topography. As described in Section 3.2.1, GSFLOW–GRASS determines surface-water flow using the GRASS GIS's efficient and accurate **r.watershed** extension, which implements a least-cost path algorithm designed to produce drainage networks that route flow in the long-range path of steepest descent regardless of the degree of local drainage integration. By using **r.watershed** alongside a set of new GRASS-GIS extensions that integrate it into the GSFLOW framework, GSFLOW–GRASS is able to automatically create a topologically correct and linked drainage network for hydrologic model simulations in settings that lack integrated drainage.

The Cannon River watershed is by far the largest of the three model implementations and thus greatly benefits from the coarsened surface and subsurface domains (Table 2). To test the robustness and efficiency of GSFLOW-GRASS over different resolution configurations, we compared the accuracy and compute time of the Cannon River example case across one order of magnitude in threshold surface drainage area (for the HRU delineation) and two orders of magnitude in MODFLOW subsurface grid cell area, starting from the base case resolution shown in Figure 7. Note that the threshold surface drainage area increase was limited by the total watershed size. Figure 7A shows that coarsening the irregular HRU resolution results in little (at finer MODFLOW resolution) to negligible (at coarser MODFLOW resolution) increase in error, compared to coarsening the rectangular MODFLOW grid. This demonstrates that the accuracy of GSFLOW-GRASS' topographically based surface discretization is well-maintained even with large-sized cells. Over the 2 orders of magnitude increase in rectangular MODFLOW grid cell sizes, errors steadily grow to about 35%, but GSFLOW-GRASS' hydrological correction step (Section 3.2.2) helped prevent even greater errors. The trade-off for accuracy is compute time: GSFLOW runtime is much more sensitive to the MODFLOW resolution than to the surface-domain resolution (Figure 7B). However, the domain builder algorithm – which requires longer to compute times than the 5.5-year GSFLOW simulation for the Cannon River – is sensitive to both the surface drainage resolution and the MODFLOW grid cell area, and it is even more sensitive to the surface drainage resolution (Figure 7C). GSFLOW-GRASS's fully automated surface drainage delineation thus allows users to overcome one of the most time-consuming obstacles to implementing integrated hydrologic models by constructing efficient and accurate irregular HRUs. The GSFLOW-GRASS toolkit makes it easy to carry out systematic GSFLOW configuration tests like this to assess model performance.

Comparisons between the simulated streamflow at the watershed outlet and corresponding observations at Welch, MN reveal that without any parameter calibrations, the model produces realistic discharge during non-peak flood times and during one of the observed peaks during July 1942 (Figure 6E). The severely over-simulated discharge in July 1943 may be evidence for a

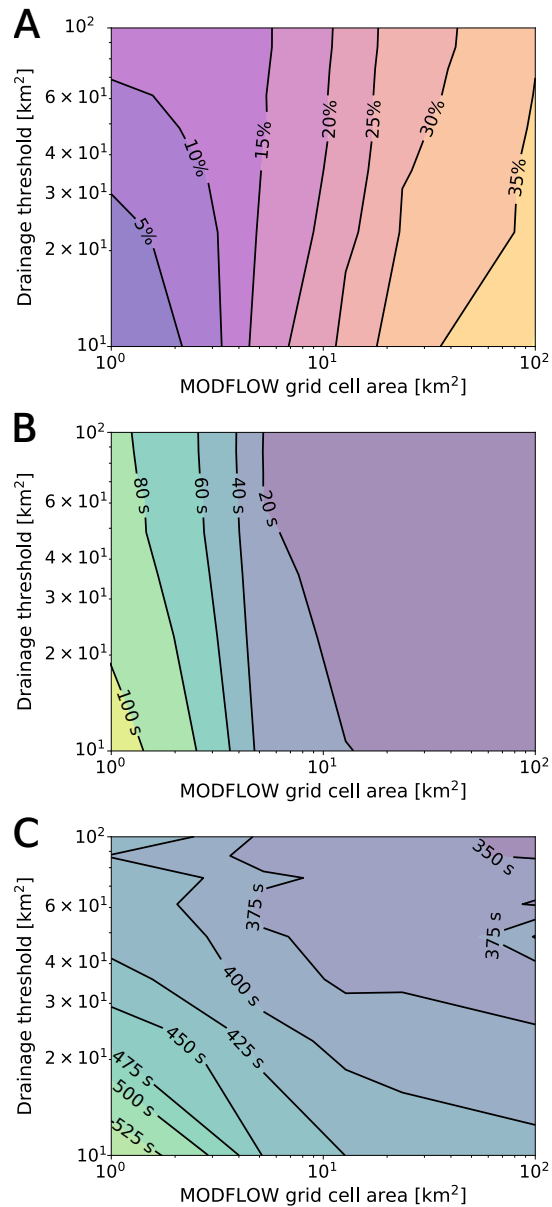



Figure 7. Changes in domain-building time, model runtime, and model output reproducibility with changes in both HRU and MODFLOW grid cell resolution. **(A)** Root-mean square error in basin discharge between each simulation and the highest-resolution simulation. **(B)** Time required to set up the input parameters and run the model. **(C)** Time required to build the domain. All computations were run on a Project Sputnik Dell XPS13 first-generation laptop with an Intel i7-4510U 2.0 GHz CPU and 8 GB of RAM running Ubuntu 18.04.

local convective summer storm system passing over the Zumbrota weather station, which is located outside of the watershed boundary. Recurring failure of the model to capture April discharge indicates that snowmelt-related parameters require adjust-

Table 3. Model implementations based on three sites serve to test GSFLOW–GRASS capabilities and demonstrate applications.

Site	GSFLOW–GRASS capabilities	Applications
Cannon River, Minnesota, USA	Two-layer hydraulic conductivity; Least-cost flow algorithm for poorly integrated drainage	Mixed agricultural—recreational watersheds; Strong temperature seasonality
Water Canyon, Santa Rosa Island, California, USA	Irregular / coastline boundaries; Hydrologically corrected coarsening; Spatially distributed hydraulic conductivity	Management of eroding hill slopes; Semi-arid climate with losing streams
Shullcas River, Junín Region, Peru	Efficient discretization of steep topography; Hydrologically corrected coarsening	Water resources in mountain catchments; Groundwater-surface water interactions under seasonally variable precipitation

Precipitation statistics from 2013-08-26 to 2016-09-29 (Shullcas); 1990-04-23 to 2017-09-27 (Water Canyon); 1938-05-12 to 1943-11-05 (Cannon). CV = coefficient of variation.

ment. The automated GSFLOW–GRASS toolkit can facilitate application of parameter-estimation approaches (e.g., Doherty, 1994; Poeter and Hill, 1998, 1999). A calibrated model can then be used to evaluate the flushing of shallow groundwater – which is susceptible to leaching from overlying agricultural plots – into the river channels during major storms, as shown in Figure 6B.

4.2 Water Canyon, Santa Rosa Island, California, USA

Santa Rosa Island is part of the Channel Islands National Park in California, USA and is characterized by mountainous topography (Clark et al., 1990). Hydrologic modeling of Santa Rosa Island has previously been performed by Jazwa et al. (2016), who applied the PIHM hydrologic model (Qu and Duffy, 2007) to the island in order to understand the relationship between prehistoric human settlement patterns and surface water availability. They reported streamflow characteristics modeled for the 19 major drainages around the island during hypothetical climate regimes. Unlike PIHM, GSFLOW–GRASS employs a regular three-dimensional groundwater grid that does not align with the irregular surface domain; this makes the integrated domain building more complicated but allows for a flexible representation of the surface-water and aquifer systems.

Here we apply the GSFLOW–GRASS toolbox to model Water Canyon (Tables 2 and 3), one of the island's many drainages. We generated the surface flow routing system with topography derived from a 3 arcsecond SRTM DEM (Farr et al., 2007) projected to a UTM coordinate system at 90 m resolution, which was coarsened for both the surface and subsurface domains (Table 2). We drove simulations shown in Figure 8 using weather data from the Western Regional Climate Center (wrcc.dri. edu).

Water Canyon is unique among the three example sites in that its outflow drains to the ocean. It therefore requires GSFLOW–GRASS to accommodate irregular boundaries (coastlines) by properly assigning boundary conditions and routing flow through them. Users identify ocean pixels by assigning NULL values to them; this causes flow routing from **r.watershed** to stop at the shoreline. To allow flow out of pour-point at the mouth of the river, the immediately downgradient MODFLOW cell can be set

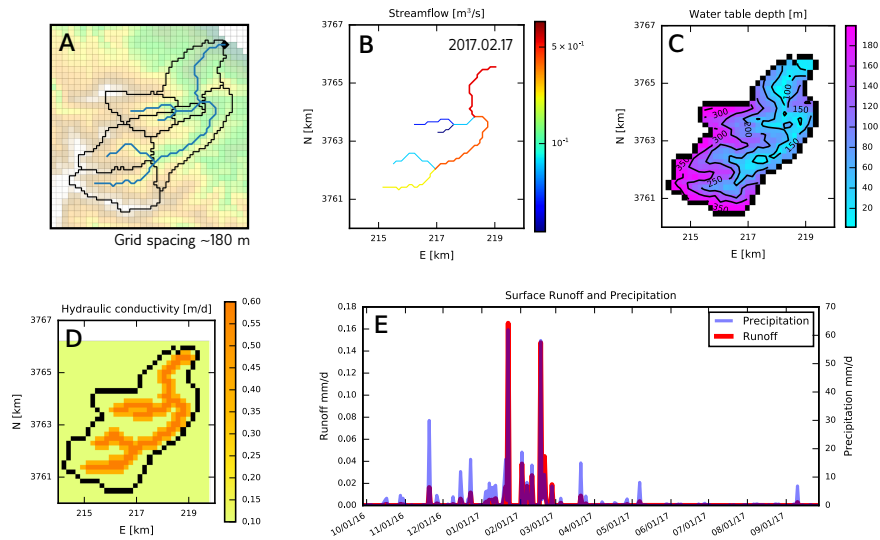



Figure 8. Model based on Water Canyon, Santa Rosa Island, California, USA. **(A)** Map with MODFLOW grid, HRU outlines, stream segments (blue), and digital elevation model. **(B)** Streamflow accumulation through the drainage network. **(C)** The modeled water table distribution with elevation contours (m.a.s.l.). **(D)** Spatially variable hydraulic conductivity structure, with hydraulic conductivity increasing near the channel to represent alluvium and colluvium. **(E)** Simulated surface runoff contributions to catchment-wide discharge compared with precipitation.

as a constant-head boundary, but this cell must be chosen carefully. The finite-difference scheme in MODFLOW dictates that the constant head boundary condition must be supplied along one of the four cardinal directions of the pour-point. Therefore, if the river flows diagonally to the sea, its constant-head boundary must be moved to the closest non-diagonal cell. **v.gsflow.grid** finds the proper constant-head boundary cell to set for the coastal case, as well as for any inland drainage case in which the
5   pour point also requires a downgradient constant-head boundary.

Losing streams such as those in the steep and semi-arid Water Canyon catchment often run dry Jazwa et al. (2016). If this causes MODFLOW cells to lose all of their water, GSFLOW will fail to numerically converge. Thus, the Water Canyon example also serves to demonstrate GSFLOW–GRASS's ability to prevent this problem by (1) incorporating MODFLOW-NWT, which uses a Newton–Raphson solver for increased stability (Niswonger et al., 2011); (2) allowing the user to specify an adequately
10 deep MODFLOW discretization in the Settings file (Section 3.1) to supply sufficient water through the dry season; and (3) hydrologically correcting the elevations of coarsened MODFLOW cells to ensure flow through the stream network (Section 3.2.2).

The Santa Rosa example demonstrates an application in which GSFLOW–GRASS can be used to investigate and manage erosion associated with hydrological conditions. The example model input script included in GSFLOW–GRASS (Section
15 3.3.3) generated a spatially distributed hydraulic conductivity field (Figure 8D) that increases near the river channels; this can

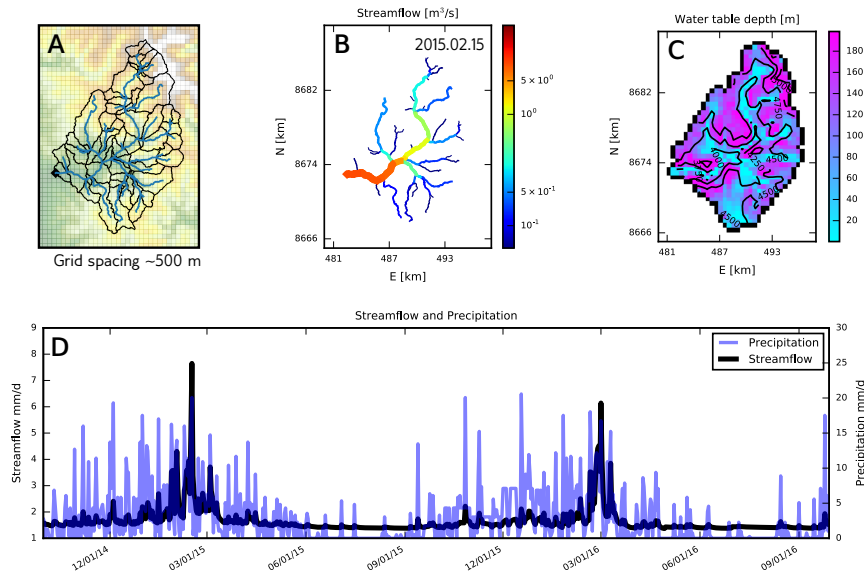



Figure 9. Model based on Río Shullcas, Peru. **(A)** Map with MODFLOW grid, HRU outlines, stream segments (blue), and digital elevation model. **(B)** Streamflow accumulation through the mountainous drainage network. **(C)** The modeled water table distribution with elevation contours (m.a.s.l.). **(D)** Seasonally variable precipitation and streamflow.

represent the transition between poorly-sorted and silt/clay-bearing hillslope soils and fluvial sands and gravels. Figure 8E demonstrates how the post-processing tools can be used to evaluate precipitation-triggered surface runoff (Figure 8E), which can denude the hillslopes (Schumann et al., 2016) and transport eroded sediment through the drainage network (Figure 8B).

4.3 Shullcas River watershed, Peru

5 The final test case is based on the Shullcas River Watershed, located in the central Peruvian Andes. Precipitation is highly seasonal, and water shortages are common during the dry season from May to September. The Huaytapallana Glacier, which supplies meltwater to the Shullcas River, is rapidly retreating (López-Moreno et al., 2014), causing concern over future water resources (Somers et al., 2018). However, in glacierized watersheds of the Peruvian Andes, a large proportion of the dry season stream discharge can be composed of groundwater (Baraer et al., 2015), driving the need to better understand groundwater-

10 surface water interactions in the catchment. Our choice of an example in the Peruvian Andes also demonstrates how our entirely open-source modeling system may be applied to problems in regions where financial limitations faced by local environmental researchers and practitioners make it difficult to use commercial software solutions.

The simple hydrologic model based on the Shullcas watershed covers a large elevation range and uses a coarsened discretization based on an ASTER nominal 30 m resolution DEM (Tachikawa et al., 2011) (Table 2). Meteorological data were

15 obtained from the Peruvian Meteorological Office (SENMAHI) online database. Located in the Andes, the Shullcas River Wa-

tershed serves as an apt testbed for examining the ability of GSFLOW–GRASS to represent surface-water–groundwater links in steep terrain. Representing flow in steep topography and narrow canyons calls for high resolution computations that can be infeasible to directly incorporate into an integrated hydrologic model. GSFLOW–GRASS solved this problem by converting high-resolution flow-routing information for Shullcas into a far smaller number of topographically defined computational surface cells and coarsened MODFLOW grid cells (see Table 2) - a strategy that the multiple-configuration Cannon River test demonstrated can generate an efficient and accurate discretization (Figure 7). The major challenge to domain coarsening further presented by Shullcas is its particular susceptibility to artificial "dams" due to its mountainous topography. These numerical artifacts occur when averaging elevations across flat valley floors and adjacent steep canyon walls, which can cause cells containing streams to be higher rather than lower than the surrounding cells on the MODFLOW grid. GSFLOW–GRASS's hydrological correction addresses this by enforcing integrated subsurface drainage (Section 3.2.2), thus preventing improper accumulation and, subsequently, lateral leakage of water behind "dammed" stream cells (Figure 3).

The Shullcas-based simulation does not include glacier melt, but spatiotemporal results in Figure 9 show that GSFLOW can be useful for evaluating the potential for groundwater to buffer surface water resources in mountainous watersheds. The model shows that essentially constant baseflow supports low but reliable discharge during the dry season (Figure 9D). The GSFLOW–GRASS post-processing visualization tools were useful for depicting the accumulation of streamflow throughout the drainage network (Figure 9B) and water table depths that were shallowest in low and flat areas (Figure 9C).

5   Conclusions

To address the need for a fully automated and freely accessible software that handles the complete workflow for implementing complex hydrologic models, we have created GSFLOW–GRASS, a bundled toolkit for the coupled surface-water and groundwater model GSFLOW, using open-source Python scripts and GRASS GIS commands. GSFLOW–GRASS allows users equipped with a DEM, precipitation and temperature data, and basic knowledge about land-surface and subsurface properties to efficiently construct watershed-scale hydrologic simulations. In order to create a robust tool that can be widely implemented over diverse hydro(geo)logic settings, we built a set of GRASS GIS extensions that automatically discretizes a topological surface-water flow network that is linked with the underlying gridded groundwater domain. Our fully automated and generalized toolbox advances the accessibility of complex hydrologic software and will thus broaden the reach of integrated hydrologic models and their usage in both scientific research and practical resource management.

We have demonstrated GSFLOW–GRASS using three diverse examples based on topographies and climates from the water-stressed Andes, Santa Rosa Island off the coast of California, USA, and the intensively farmed Upper Midwest region of the United States. The results show that the new and automated GRASS GIS extensions can automatically and consistently build topologically complete linked surface and subsurface flow domains in settings that are typically challenging for standard GIS tools, including low-relief terrains that lack integrated drainage, irregular coastal boundaries, and steep topographies. Although uncalibrated, these examples further demonstrate that GSFLOW–GRASS is a flexible tool for investigating the role of

groundwater-surface water interactions in imposing possible water-quality threats in agricultural and recreational watersheds, controlling runoff in erosion-prone landscapes, and modulating dry-season discharge.

We designed GSFLOW–GRASS to strike a balance between direct "out-of-the-box" functionality and full flexibility for customizing model runs. A default implementation can be launched with no programming required by the user to readily produce

5 preliminary uncalibrated simulations that can serve as a springboard for further model-parameter adjustment through the fully commented toolkit scripts. A key feature of GSFLOW–GRASS is its use of all open-source software, enabling users anywhere to apply GSFLOW. We believe that the open-source platform will facilitate future toolbox enhancements through efforts by not only the original GSFLOW–GRASS developer team, but also new model users. We envision a number of new capabilities to tackle the grand challenge of handling spatial heterogeneity in integrated hydrologic models. Higher resolution land-surface

10 variability could be achieved by further subdividing sub-basins according to vegetation, soil type, or other geographic features to produce HRUs. Obtaining spatially variable information can be facilitated by linking GSFLOW–GRASS to existing regional to international databases for meteorology, soil and geologic properties, and land cover. Further calibration of spatially distributed parameters can be carried out by directly setting up GSFLOW–GRASS with a flexible inverse modeling code (e.g., Doherty, 1994; Poeter and Hill, 1998, 1999). It is our hope that with its generalized form and open-access, GSFLOW–GRASS

15 can become a community tool that continues to grow to better solve hydrologic and water resources problems of both scientific and general management concerns.

Code availability. The version of GSFLOW–GRASS used for this paper is available at https://github.com/UMN-Hydro/GSFLOW--GRASS/ releases. Updated versions of our code are downloadable directly from the UMN-Hydro repository on GitHub, at https://github.com/ UMN-Hydro/GSFLOW--GRASS. The user's manual is available as the README.md file in the repository. The GSFLOW executable

20 and source code are available in the UMN-Hydro repository https://github.com/UMN-Hydro/GSFLOW-1.2.0 and from the USGS website https://water.usgs.gov/ogw/gsflow/. GRASS GIS 7.3+ is available from https://grass.osgeo.org/download/software/.

Author contributions. G.-H. C. Ng and A. D. Wickert contributed equally to this work; they together conceived of the project idea, wrote the majority of the paper, and built the visualization scripts. G.-H. C. Ng wrote the input file builder and plotting scripts. A. D. Wickert built the GRASS GIS extensions and integrated them with Ng's workflow. L. D. Somers and C. Cronkite-Ratcliff tested the code during its

25 development and contributed their study cases. L. Saberi helped code the input-file builder scripts. R. Niswonger provided advice and support from the U.S. Geological Survey team that develops GSFLOW. J. M. McKenzie provided support for the Shullcas test case.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. The authors acknowledge all contributors to MODFLOW, PRMS, and GSFLOW, without whom this work would not be possible. Helpful comments from three anonymous referees guided manuscript improvements.

References

Adresen, J., Hilberg, S., and Kunkel, K.: Historical climate and climate trends in the Midwestern United States, Climate Change in the Midwest: A Synthesis Report for the National Climate Assessment, pp. 8–36, 2014.

Ajami, N. K., Gupta, H., Wagener, T., and Sorooshian, S.: Calibration of a semi-distributed hydrologic model for streamflow estimation along a river system, Journal of Hydrology, 298, 112–135, https://doi.org/10.1016/j.jhydrol.2004.03.033, 2004.

Arge, L., Chase, J. S., Halpin, P., Toma, L., Vitter, J. S., Urban, D., and Wickremesinghe, R.: Efficient flow computation on massive grid terrain datasets, GeoInformatica, 7, 283–313, https://doi.org/10.1023/A:1025526421410, 2003.

Arnold, J. G. and Fohrer, N.: SWAT2000: Current capabilities and research opportunities in applied watershed modelling, Hydrological Processes, 19, 563–572, https://doi.org/10.1002/hyp.5611, 2005.

Arnold, J. G., Moriasi, D. N., Gassman, P. W., Abbaspour, K. C., and White, M. W.: SWAT: Model use, calibration, and validation, Biological Systems Engineering, 55, 1491 – 1508, 2012.

Bandaragoda, C., Tarboton, D. G., and Woods, R.: Application of TOPNET in the distributed model intercomparison project, Journal of Hydrology, 298, 178–201, https://doi.org/10.1016/j.jhydrol.2004.03.038, http://linkinghub.elsevier.com/retrieve/pii/S0022169404002446, 2004.

Baraer, M., Mckenzie, J., Mark, B. G., Gordon, R., Bury, J., Condom, T., Gomez, J., Knox, S., and Fortner, S. K.: Contribution of groundwater to the outflow from ungauged glacierized catchments: A multi-site study in the tropical Cordillera Blanca, Peru, Hydrological Processes, 29, 2561–2581, https://doi.org/10.1002/hyp.10386, 2015.

Barnes, R., Lehman, C., and Mulla, D.: Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models, Computers and Geosciences, 62, 117–127, https://doi.org/10.1016/j.cageo.2013.04.024, http://dx.doi.org/10.1016/j.cageo.2013.04.024, 2014.

Beven, K.: A manifesto for the equifinality thesis, Journal of Hydrology, 320, 18–36, https://doi.org/10.1016/j.jhydrol.2005.07.007, 2006.

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

Bhowmik, A. K., Metz, M., and Schäfer, R. B.: An automated, objective and open source tool for stream threshold selection and upstream riparian corridor delineation, Environmental Modelling and Software, 63, 240–250, https://doi.org/10.1016/j.envsoft.2014.10.017, 2015.

Bird, M. I., O'Grady, D., and Ulm, S.: Humans, water, and the colonization of Australia, Proceedings of the National Academy of Sciences, 113, 11 477–11 482, https://doi.org/10.1073/pnas.1608470113, http://www.pnas.org/lookup/doi/10.1073/pnas.1608470113, 2016.

Braun, J. and Willett, S. D.: A very efficient O(n), implicit and parallel method to solve the stream power equation governing fluvial incision and landscape evolution, Geomorphology, 180-181, 170–179, https://doi.org/10.1016/j.geomorph.2012.10.008, http://dx.doi.org/10.1016/j.geomorph.2012.10.008http://linkinghub.elsevier.com/retrieve/pii/S0169555X12004618, 2013.

Butts, M. and Graham, D.: Flexible Integrated Watershed Modeling with MIKE SHE, in: Watershed Models, pp. 245–271, CRC Press, https://doi.org/10.1201/9781420037432.ch10, http://www.crcnetbase.com/doi/10.1201/9781420037432.ch10, 2005.

Clark, R. A., Halvorson, W. L., Sawdo, A. A., and Danielsen, K. C.: Plant communities of Santa Rosa Island, Channel Islands National Park, vol. Technical, University of California, Davis, California, 1990.

Czuba, J. A. and Foufoula-Georgiou, E.: A network-based framework for identifying potential synchronizations and amplifications of sediment delivery in river basins, Water Resources Research, 50, 3826–3851, https://doi.org/10.1002/2013WR014227, 2014.

Doherty, J.: PEST: A Unique Computer Program for Model-independent Parameter Optimisation, in: Water Down Under 94: Groundwater/Surface Hydrology Common Interest Papers; Preprints of Papers, p. 551, http://search.informit.com.au/documentSummary;dn=752715546665009;res=IELENG, 1994.

Essaid, H. I. and Hill, B. R.: Watershed-scale modeling of streamflow change in incised montane meadows, Water Resources Research, 50, 2657–2678, https://doi.org/10.1002/2013WR014420, 2014.

Farr, T. G., Rosen, P. A., Caro, E., Crippen, R., Duren, R., Hensley, S., Kobrick, M., Paller, M., Rodriguez, E., Roth, L., Seal, D., Shaffer, S., Shimada, J., Umland, J., Werner, M., Oskin, M., Burbank, D., and Alsdorf, D.: The Shuttle Radar Topography Mission, Reviews of Geophysics, 45, RG2004, https://doi.org/10.1029/2005RG000183, http://doi.wiley.com/10.1029/2005RG000183, 2007.

Feldman, A. D.: Hydrologic modeling system HEC-HMS, Technical Reference Manual, Technical Reference Manual, p. 145, https://doi.org/CDP-74B, http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Hydrologic+Modeling+System+HEC-HMS#7, 2000.

Galeone, D. G., Risser, D. W., Eicholtz, L. W., and Hoffman, S. A.: Water Quality and Quantity and Simulated Surface-Water and Groundwater Flow in the Laurel Hill Creek Basin, Southwestern Pennsylvania, 1991-2007, vol. 2016-5082 of *Scientific Investigations Report*, U.S. Geological Survey, https://doi.org/10.3133/sir20165082, 2016.

Gallagher, M. and Doherty, J.: Parameter estimation and uncertainty analysis for a watershed model, Environmental Modelling & Software, 22, 1000–1020, https://doi.org/10.1016/j.envsoft.2006.06.007, http://linkinghub.elsevier.com/retrieve/pii/S1364815206001629, 2007.

Gannett, M., Lite, K. J., Risley, J., Pischel, E., and La Marche, J.: Simulation of Groundwater and Surface-Water Flow in the Upper Deschutes Basin, Oregon, vol. 2017–5097 of *Scientific Investigations Report*, U.S. Geological Survey, https://doi.org/10.3133/sir20175097, 2017.

Gardner, M. A., Morton, C. G., Huntington, J. L., Niswonger, R. G., and Henson, W. R.: Input data processing tools for the integrated hydrologic model GSFLOW, Environmental Modelling & Software, https://doi.org/10.1016/j.envsoft.2018.07.020, https://linkinghub.elsevier.com/retrieve/pii/S1364815217311908, 2018.

Harbaugh, A. W.: MODFLOW-2005 , The U . S . Geological Survey Modular Ground-Water Model — the Ground-Water Flow Process, in: Book 6. Modeling techniques, Section A. Ground Water, vol. U.S. Geolo, chap. 16, p. 253, U.S. Geological Survey, Reston, Virginia, USA, https://doi.org/U.S. Geological Survey Techniques and Methods 6-A16, 2005.

Hassan, S. T., Lubczynski, M. W., Niswonger, R. G., and Su, Z.: Surface–groundwater interactions in hard rocks in Sardon Catchment of western Spain: An integrated modeling approach, Journal of Hydrology, 517, 390–410, https://doi.org/10.1016/j.jhydrol.2014.05.026, http://dx.doi.org/10.1016/j.jhydrol.2014.05.026http://linkinghub.elsevier.com/retrieve/pii/S0022169414003904, 2014.

Heckmann, T., Schwanghart, W., and Phillips, J. D.: Graph theory-Recent developments of its application in geomorphology, Geomorphology, 243, 130–146, https://doi.org/10.1016/j.geomorph.2014.12.024, http://dx.doi.org/10.1016/j.geomorph.2014.12.024, 2014.

Hofierka, J., Mitášová, H., and Neteler, M.: Chapter 17 Geomorphometry in GRASS GIS, in: Developments in Soil Science, vol. 33, chap. 17, pp. 387–410, https://doi.org/10.1016/S0166-2481(08)00017-2, http://linkinghub.elsevier.com/retrieve/pii/S0166248108000172, 2009.

Hsieh, P. A. and Winston, R. B.: User's Guide To Model Viewer, a Program for Three-Dimensional Visualization of Ground-Water Model Results, vol. 02-106 of *Open-File Report*, U.S. Geological Survey, Menlo Park, CA, 2002.

Hunt, R. J., Walker, J. F., Selbig, W. R., Westenbroek, S. M., and Regan, R. S.: Simulation of Climate - Change effects on streamflow, Lake water budgets, and stream temperature using GSFLOW and SNTEMP, Trout Lake Watershed, Wisconsin, USGS Scientific Investigations Report., pp. 2013–5159, 2013.

Hunter, J. D.: Matplotlib: A 2D graphics environment, Computing in Science and Engineering, 9, 99–104, https://doi.org/10.1109/MCSE.2007.55, 2007.

Jasiewicz, J. and Metz, M.: A new GRASS GIS toolkit for Hortonian analysis of drainage networks, Computers and Geosciences, 37, 1162–1173, https://doi.org/10.1016/j.cageo.2011.03.003, 2011.

Jazwa, C. S., Duffy, C. J., Leonard, L., and Kennett, D. J.: Hydrological Modeling and Prehistoric Settlement on Santa Rosa Island, California, USA, Geoarchaeology, 31, 101–120, https://doi.org/10.1002/gea.21532, 2016.

Jenson, S. K. and Domingue, J. O.: Extracting topographic structure from digital elevation data for geographic information system analysis, Photogrammetric Engineering and Remote Sensing, 54, 1593–1600, https://doi.org/0099-1112/88/5411-1593$02.25/0, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.138.6487&rep=rep1&type=pdf, 1988.

Kinner, D., Mitasova, H., Stallard, R., Harmon, R. S., and Toma, L.: GIS-Based Stream Network Analysis for the Upper Río Chagres Basin, Panama, in: The Río Chagres, Panama: A Multidisciplinary Profile of a Tropical Watershed, edited by Harmon, R. S., pp. 83–95, Springer-Verlag, Berlin/Heidelberg, https://doi.org/10.1007/1-4020-3297-8_6, http://link.springer.com/10.1007/1-4020-3297-8_6, 2005.

Kreiling, R. M. and Houser, J. N.: Long-term decreases in phosphorus and suspended solids, but not nitrogen, in six upper Mississippi River tributaries, 1991–2014, Environmental Monitoring and Assessment, 188, https://doi.org/10.1007/s10661-016-5464-3, http://dx.doi.org/10.1007/s10661-016-5464-3, 2016.

Leavesley, G. H., Lichty, R., Troutman, B., and Saindon, L.: Precipitation-Runoff Modeling System: User's Manual, vol. Water Reso, U.S. Geological Survey, Denver, Colorado, USA, 1983.

Leavesley, G. H., Restrepo, P., Markstrom, S., Dixon, M., and Stannard, L.: The Modular Modeling System (MMS): User´s Manual, vol. 1, 1996.

Leonard, L. and Duffy, C. J.: Essential Terrestrial Variable data workflows for distributed water resources modeling, Environmental Modelling & Software, 50, 85–96, https://doi.org/10.1016/j.envsoft.2013.09.003, http://linkinghub.elsevier.com/retrieve/pii/S1364815213001941, 2013.

Leopold, L. B. and Maddock, T.: The hydraulic geometry of stream channels and some physiographic implications, Professional Paper, United States Geological Survey, Washington, D.C., 1953.

LeVeque, R. J.: Finite volume methods for hyperbolic problems, vol. 31, Cambridge university press, 2002.

López-Moreno, J., Fontaneda, S., Bazo, J., Revuelto, J., Azorin-Molina, C., Valero-Garcés, B., Morán-Tejeda, E., Vicente-Serrano, S., Zubieta, R., and Alejo-Cochachín, J.: Recent glacier retreat and climate trends in Cordillera Huaytapallana, Peru, Global and Planetary Change, 112, 1–11, https://doi.org/10.1016/j.gloplacha.2013.10.010, http://linkinghub.elsevier.com/retrieve/pii/S0921818113002385, 2014.

Luzio, M. D., Arnold, J. G., and Srinivasan, R.: A GIS Coupled hydrological model system for the watershed assessment of agricultural nonpoint and point sources of polution, Transactions in GIS, 8, 113–136, https://doi.org/10.1111/j.1467-9671.2004.00170.x, 2006.

Magalhães, S. V. G., Andrade, M. V. A., Franklin, W. R., and Pena, G. C.: A linear time algorithm to compute the drainage network on grid terrains, Journal of Hydroinformatics, 16, 1227, https://doi.org/10.2166/hydro.2013.068, http://jh.iwaponline.com/cgi/doi/10.2166/hydro.2013.068, 2014.

Maidment, D. R. and Morehouse, S.: Arc Hydro: GIS for water resources, vol. 1, ESRI, Inc., 2002.

Markstrom, S. L., Niswonger, R. G., Regan, R. S., Prudic, D. E., and Barlow, P. M.: GSFLOW—Coupled Ground-Water and Surface-Water Flow Model Based on the Integration of the Precipitation-Runoff Modeling System (PRMS) and the Modular Ground-Water Flow Model (MODFLOW-2005), U.S. Geological Survey, p. 240, https://doi.org/10.13140/2.1.2741.9202, http://pubs.er.usgs.gov/publication/tm6D1, 2008.

Markstrom, S. L., Regan, R. S., Hay, L. E., Viger, R. J., Webb, R. M. T., Payn, R. A., and LaFontaine, J. H.: PRMS-IV, the Precipitation-Runoff Modeling System, Version 4, U.S. Geological Survey, Reston, Virginia, USA, https://doi.org/http://dx.doi.org/10.3133/tm6B7, 2015.

Maxwell, R., Putti, M., Meyerhoff, S., Delfs, J., Ferguson, I. M., Ivanov, V., Kim, J., Kolditz, O., Kollet, S. J., Kumar, M., Lopez, S., Niu, J., Paniconi, C., Park, Y., Phanikumar, M., Shen, C., Sudicky, E. A., and Sulis, M.: Water Resources Research, Water resources research, 50, 1531–1549, https://doi.org/10.1002/2013WR013725.Received, 2014.

Maxwell, R. M., Kollet, S. J., Smith, S. G., Woodward, C. S., Falgout, R. D., Ferguson, I. M., Engdahl, N., Condon, L. E., Hector, B., Lopez, S., Bearup, L., Jefferson, J., Collins, C., Graaf, I. D., Pribulick, C., Baldwin, C., Bosl, W. J., Hornung, R., and Ashby, S.: ParFlow User's Manual, Integrated Ground- Water Modeling Center Report GWMI 2016-01, 2017.

Metz, M., Mitasova, H., and Harmon, R. S.: Efficient extraction of drainage networks from massive, radar-based elevation models with least cost path search, Hydrology and Earth System Sciences, 15, 667–678, https://doi.org/10.5194/hess-15-667-2011, http://www.hydrol-earth-syst-sci.net/15/667/2011/, 2011.

Miliaresis, G. and Delikaraoglou, D.: Effects of percent tree canopy density and DEM misregistration on SRTM/NED vegetation height estimates, Remote Sensing, 1, 36–49, https://doi.org/10.3390/rs1020036, 2009.

Mitasova, H., Mitas, L., Brown, W. M., Gerdes, D. P., Kosinovsky, R., and Baker, T.: Modelling spatially and temporally distributed phenomena: New methods and tools for GRASS GIS, International Journal of Geographical Information Systems, 9, 433–446, https://doi.org/10.1080/02693799508902048, 1995.

Neitsch, S., Arnold, J., Kiniry, J., Srinivasan, R., and Williams, J.: Soil and Water Assessment Tool User's Manual, vol. TR-192, Texas Water Resources Institute, College Station, Texas, USA, http://swat.tamu.edu/media/1294/swatuserman.pdf, 2002.

Neteler, M. and Mitasova, H.: Open Source GIS: A GRASS GIS Approach, Springer, New York, New York, USA, third edit edn., 2008.

Neteler, M., Beaudette, D., Cavallini, P., Lami, L., and Cepicky, J.: GRASS GIS, Open Source Approaches in Spatial Data Handling, 2, 171–199, https://doi.org/10.1007/978-3-540-74831-1_9, http://dx.doi.org/10.1007/978-3-540-74831-1_9, 2008.

Neteler, M., Bowman, M. H., Landa, M., and Metz, M.: GRASS GIS: A multi-purpose open source GIS, Environmental Modelling & Software, 31, 124–130, https://doi.org/10.1016/j.envsoft.2011.11.014, http://dx.doi.org/10.1016/j.envsoft.2011.11.014http://linkinghub.elsevier.com/retrieve/pii/S1364815211002775, 2012.

Niswonger, R. G., Panday, S., and Motomu, I.: MODFLOW-NWT, A Newton Formulation for MODFLOW-2005, vol. Techniques, U.S. Geological Survey, Reston, Virginia, USA, 2011.

Pal, J. S., Giorgi, F., Bi, X., Elguindi, N., Solmon, F., Gao, X., Rauscher, S. A., Francisco, R., Zakey, A., Winter, J., Ashfaq, M., Syed, F. S., Bell, J. L., Differbaugh, N. S., Karmacharya, J., Konari, A., Martinez, D., Da Rocha, R. P., Sloan, L. C., and Steiner, A. L.: Regional climate modeling for the developing world: The ICTP RegCM3 and RegCNET, Bulletin of the American Meteorological Society, 88, 1395–1409, https://doi.org/10.1175/BAMS-88-9-1395, 2007.

Patterson, C. J. and Hobbs, H. C.: Surficial geology, in: County Atlas C-9: Geologic atlas of Rice County, edited by Hobbs, H. C., vol. pl. 3 of *1:100,000*, Minnesota Geological Survey, Saint Paul, Minnesota, USA, 1995.

Pérez, F. and Granger, B. E.: IPython: A system for interactive scientific computing, Computing in Science and Engineering, 9, 21–29, https://doi.org/10.1109/MCSE.2007.53, 2007.

Poeter, E. P. and Hill, M. C.: Documentation of UCODE, A Computer Code for Universal Inverse Modeling, USGS Water Resources Investigations Report 98-4080, vol. Water-Reso, U.S. Geological Survey, Denver, Colorado, USA, 1998.

Poeter, E. P. and Hill, M. C.: UCODE, a computer code for universal inverse modeling, Computers and Geosciences, 25, 457–462, https://doi.org/10.1016/S0098-3004(98)00149-6, 1999.

QGIS Development Team: QGIS Geographic Information System, http://qgis.osgeo.org, 2013.

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

Razavi, S. and Gupta, H. V.: What do we mean by sensitivity analysis? The need for comprehensive characterization of "global" sensitivity in Earth and Environmental systems models, Water Resources Research, 51, 3070–3092, https://doi.org/10.1002/2014WR016527.Received, 2015.

Reed, S., Koren, V., Smith, M., Zhang, Z., Moreda, F., and Seo, D. J.: Overall distributed model intercomparison project results, Journal of Hydrology, 298, 27–60, https://doi.org/10.1016/j.jhydrol.2004.03.031, 2004.

Regan, R. S., Niswonger, R. G., Markstrom, S. L., and Barlow, P. M.: Documentation of a restart option for the U.S. Geological Survey coupled groundwater and surface-water flow (GSFLOW) model, vol. 6-D3, U.S. Geological Survey, Reston, Virginia, USA, 2015.

Reilly, T. E.: System and Boundary Conceptualization in Ground-Water Flow Simulation, in: Techniques of Water-Resources Investigations of the United States Geological Survey, Book 3, Applications of Hydraulics, chap. B8, p. 38, U.S. Geological Survey, 2001.

Reilly, T. E. and Harbaugh, A. W.: Guidelines for Evaluating Ground-Water Flow Models, vol. 2004-5038, U.S. Geological Survey, 2004.

Rossi, M. and Reichenbach, P.: LAND-SE: A software for statistically based landslide susceptibility zonation, version 1.0, Geoscientific Model Development, 9, 3533–3543, https://doi.org/10.5194/gmd-9-3533-2016, 2016.

Sangireddy, H., Stark, C. P., Kladzyk, A., and Passalacqua, P.: GeoNet: An open source software for the automatic and objective extraction of channel heads, channel network, and channel morphology from high resolution topography data, Environmental Modelling & Software, 83, 58–73, https://doi.org/10.1016/j.envsoft.2016.04.026, http://linkinghub.elsevier.com/retrieve/pii/S1364815216301219, 2016.

Schumann, R. R., Pigati, J. S., and McGeehin, J. P.: Fluvial system response to late Pleistocene-Holocene sea-level change on Santa Rosa Island, Channel Islands National Park, California, Geomorphology, 268, 322–340, https://doi.org/10.1016/j.geomorph.2016.05.033, http://dx.doi.org/10.1016/j.geomorph.2016.05.033, 2016.

Schwanghart, W. and Scherler, D.: Short Communication: TopoToolbox 2 – MATLAB-based software for topographic analysis and modeling in Earth surface sciences, Earth Surface Dynamics, 2, 1–7, https://doi.org/10.5194/esurf-2-1-2014, http://www.earth-surf-dynam.net/2/1/2014/, 2014.

Shapiro, M. and Westervelt, J.: r. mapcalc: An algebra for GIS and image processing, vol. USACERL-TR, Construction Engineering Research Lab, Champaign, Illinois, USA, 1994.

Simunek, J., Sejna, M., Saito, H., Sakai, M., and van Genuchten, M.: "The HYDRUS-1D software package for simulating the one-dimensional movement of water, heat, and multiple solutes in variably-saturated media. Version 4.08. HYDRUS Softw. Ser. 3., Tech. Rep. January, 2009.

Somers, L. D., McKenzie, J. M., Zipper, S. C., Mark, B. G., Lagos, P., and Baraer, M.: Does hillslope trenching enhance groundwater recharge and baseflow in the Peruvian Andes?, Hydrological Processes, 32, 318–331, https://doi.org/10.1002/hyp.11423, 2018.

Song, X., Zhang, J., Zhan, C., Xuan, Y., Ye, M., and Xu, C.: Global sensitivity analysis in hydrological modeling: Review of concepts, methods, theoretical framework, and applications, Journal of Hydrology, 523, 739–757, https://doi.org/10.1016/j.jhydrol.2015.02.013, http://dx.doi.org/10.1016/j.jhydrol.2015.02.013, 2015.

Srinivasan, R. and Arnold, J. G.: Integration of a basin-scale water quality model with GIS, Journal of the American Water Resources Association, 30, 453–462, https://doi.org/10.1111/j.1752-1688.1994.tb03304.x, http://doi.wiley.com/10.1111/j.1752-1688.1994.tb03304.x, 1994.

Steenberg, J. R., Tipping, R. G., and Runkel, A. C.: Geologic controls on groundwater and surface water flow in southeastern Minnesota and
5   its impact on nitrate concentrations in streams, Minnesota Geological Survey Open File Report, p. 154, 2013.

Surfleet, C. G. and Tullos, D.: Uncertainty in hydrologic modelling for estimating hydrologic response due to climate change (Santiam River, Oregon), Hydrological Processes, 27, 3560–3576, https://doi.org/10.1002/hyp.9485, http://doi.wiley.com/10.1002/hyp.9485, 2013.

Tachikawa, T., Kaku, M., Iwasaki, A., Gesch, D., Oimoen, M., Zhang, Z., Danielson, J., Krieger, T., Curtis, B., Haase, J., Abrams, M., Crippen, R., Carabajal, C., and ASTER GDEM Validation Team: ASTER Global Digital Elevation Model Version
10   2 – Summary of Validation Results, Tech. rep., NASA Earth Resources Observation and Science (EROS) Center, Sioux Falls, South Dakota, USA, https://doi.org/10.1017/CBO9781107415324.004, http://www.jspacesystems.or.jp/ersdac/GDEM/ver2Validation/Summary_GDEM2_validation_report_final.pdf, 2011.

Tejedor, A., Longjas, A., Zaliapin, I., and Foufoula-Georgiou, E.: Delta channel networks: 1. A graph-theoretic approach for studying connectivity and steady state transport on deltaic surfaces, Water Resources Research, 51, 3998–4018, https://doi.org/10.1002/2014WR016577,
15   http://doi.wiley.com/10.1002/2014WR016577, 2015.

Tian, Y., Zheng, Y., Wu, B., Wu, X., Liu, J., and Zheng, C.: Modeling surface water-groundwater interaction in arid and semi-arid regions with intensive agriculture, Environmental Modelling and Software, 63, 170–184, https://doi.org/10.1016/j.envsoft.2014.10.011, 2015.

Tian, Y., Zheng, Y., and Zheng, C.: Development of a visualization tool for integrated surface water-groundwater modeling, Computers and Geosciences, 86, 1–14, https://doi.org/10.1016/j.cageo.2015.09.019, http://dx.doi.org/10.1016/j.cageo.2015.09.019, 2016.

20   Tipping, R. G.: Subsurface recharge and surface infiltration, in: Geologic Atlas of Scott County, Minnesota, Minnesota Geological Survey Atlas Series, 2006.

Viger, R. J. and Leavesley, G. H.: The GIS Weasel User's Manual, U.S. Geological Survey, 2007.

Vivoni, E. R., Ivanov, V. Y., Bras, R. L., and Entekhabi, D.: Generation of Triangulated Irregular Networks Based on Hydrological Similarity, Journal of Hydrologic Engineering, 9, 288–302, https://doi.org/10.1061/(ASCE)1084-0699(2004)9:4(288), 2004.

25   Wang, D., Liu, Y., and Kumar, M.: Using nested discretization for a detailed yet computationally efficient simulation of local hydrology in a distributed hydrologic model, Scientific Reports, 8, 1–13, https://doi.org/10.1038/s41598-018-24122-7, http://dx.doi.org/10.1038/s41598-018-24122-7, 2018.

Waterloo Hydrogeologic Inc.: Visual MODFLOW 2011.1 User's Manual: For Professional Applications in Three-Dimensional Groundwater Flow and Contaminant Transport Modeling, Waterloo Hydrologic, Inc., Waterloo, Ontario, Canada, http://trials.swstechnology.com/
30   software/Visual_MODFLOW/2011/Manuals_and_Guides/VMOD-2011.1_Manual.pdf, 2011.

Wickert, A. D., Martin, J. M., Tal, M., Kim, W., Sheets, B., Paola, C., Paola, S. C., Sheets, B., Paola, C., Paola, S. C., Sheets, B., and Paola, C.: River channel lateral mobility: Metrics, time scales, and controls, Journal of Geophysical Research: Earth Surface, 118, 396–412, https://doi.org/10.1029/2012JF002386, http://doi.wiley.com/10.1029/2012JF002386, 2013.

Winston, R. B.: Graphical User Interface for MODFLOW, Version 4, vol. 00-315 of *Open-File Report*, U.S. Geological Survey, Reston,
35   Virginia, USA, 2000.

Winston, R. B.: ModelMuse: A Graphical User Interface for MODFLOW-2005 and PHAST, in: Section A, Ground Water—Book 6, Modeling Techniques, vol. 6-A29 of *Techniques and Methods*, chap. 29, p. 52 p, U.S. Geological Survey, Reston, Virginia, USA, http://pubs.usgs.gov/tm/tm6A29/tm6A29.pdf, 2009.

Winter, T. C., Harvey, J. W., Franke, O. L., and Alley, W. M.: Ground water and surface water: A single resource, U.S. Government Printing Office, Denver, Colorado, USA, 1998.

Zambelli, P., Gebbert, S., and Ciolli, M.: Pygrass: An Object Oriented Python Application Programming Interface (API) for Geographic Resources Analysis Support System (GRASS) Geographic Information System (GIS), ISPRS International Journal of Geo-Information, 2, 201–219, https://doi.org/10.3390/ijgi2010201, http://www.mdpi.com/2220-9964/2/1/201/, 2013.

Ŝúri, M. and Hofierka, J.: A new GIS-based solar radiation model and its application to photovoltaic assessments, Transactions in GIS, 8, 175–190, https://doi.org/10.1111/j.1467-9671.2004.00174.x, 2004.