Environmental Modelling and Software 109 (2018) 41–53

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

Input data processing tools for the integrated hydrologic model GSFLOW

Murphy A. Gardner[a,*], Charles G. Morton[b], Justin L. Huntington[b], Richard G. Niswonger[c], Wesley R. Henson[d]

[a] U.S. Geological Survey, Carson City, NV, United States
[b] Desert Research Institute, Reno, NV, United States
[c] U.S. Geological Survey, Menlo Park, CA, United States
[d] U.S. Geological Survey, San Diego, CA, United States

A B S T R A C T

Integrated hydrologic modeling (IHM) encompasses a vast number of processes and specifications, variable in time and space, and development of models can be arduous. Model input construction techniques have not been formalized or made easily reproducible. Creating the input files for integrated hydrologic models requires complex GIS processing of raster and vector datasets from various sources. Developing stream network topology that is consistent with the model grid-scale digital elevation model (DEM) is important for robust simulation of surface water and groundwater exchanges. Distribution of meteorological data over the model domain is difficult in complex terrain at the model-grid scale, but is necessary for realistic simulations. As model development requires extensive GIS and computer programming expertise, the use of IHMs has mostly been limited to research groups with available financial, human, and technical resources. Here we present a series of open-source Python scripts that are combined with ESRI ArcGIS to provide a formalized technique for the parameterization and development of inputs for the readily available IHM called GSFLOW. This Python toolkit automates many of the necessary and laborious processes of parameterization, including stream network development, land coverages, and meteorological distribution over the model domain. The final products of the toolkit are PRMS ready Parameter Files, along with several input parameters for a MODFLOW model, including input for the Streamflow Routing Package. A demonstration of the toolkit is provided to illustrate its capabilities.

Program and computing requirements

Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Mandatory software:

- ArcGIS 10.1 or newer (will include Python 2.7 along with the ArcPy package http://pro.arcgis.com/en/pro-app/arcpy/get-started/what-is-arcpy-.htm)
- Cascade Routing Tool (CRT) executable (version 1.3.1) http://water.usgs.gov/ogw/CRT/
- GSFLOW 1.1.6 or newer http://water.usgs.gov/ogw/gsflow/#downloads

Optional software:

- USDA Soil Data Viewer http://www.nrcs.usda.gov/wps/portal/nrcs/detail/soils/survey/geo/?cid=nrcs142p2_053614

Software availability

Version 1 release on GitHub (https://github.com/gsflow/gsflow-arcpy). Software download includes a user guide and example model.

1. Introduction

Water resource managers and researchers require greater understanding of the connectivity between groundwater and surface water to effectively manage these resources. Integrated Hydrologic Models (IHMs) can provide important information about water resources and are often used as decision support tools for resource management (Laniak et al., 2013). Construction of IHM inputs are not well formalized or automated, making reproducibility very difficult. A wider acceptance and distribution of process-based approaches requires improved model visualization tools and streamlined approaches for model setup, execution, and analysis (Fatichi et al., 2016). Various techniques have been developed to create drainage networks, but none have the benefit of a fully automated and topologically consistent approach that is applicable for simulating exchanges between streams and groundwater in grid-based IHMs applied to developed river basins.

* Corresponding author.
E-mail address: mgardner@usgs.gov (M.A. Gardner).

https://doi.org/10.1016/j.envsoft.2018.07.020
Received 14 November 2017; Received in revised form 18 June 2018; Accepted 30 July 2018
Available online 09 August 2018
1364-8152/ Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/BY-NC-ND/4.0/).

The progression of IHMs has spurred research and development for improving data processing and graphical user interfaces (Band, 1986; Gruber and Peckham, 2009; Metz et al., 2011; Wilson, 2012; Tian et al., 2016; Bhatt et al., 2008; Ng et al., 2018). Advancements in GIS technology and remote sensing have provided the ability to highly parameterize systems of interest, but not without complex data processing (Metz et al., 2011; Jasiewicz and Metz, 2011). Despite these advancements in GIS data processing, previous works have not focused on additional complexities related to simulating surface water and groundwater exchanges, and input data development for fine-resolution large-scale hydrologic models is still a difficult problem that discourages and limits the use of IHMs. Application of approaches strictly developed for surface networks to develop input for IHMs can lead to erroneous input that compromises a model's usefulness (Clark et al., 2015). Just the near-surface component of GSFLOW requires input data of approximately 50 global parameters and 35 model-cell-specific parameters that may be derived from large geospatial datasets (Markstrom et al., 2008).

Creating input files for IHMs requires complex GIS processing of raster and vector datasets from various sources (e.g., Wilson, 2012). Viger and Leavesley (2007) provide the basic steps for developing input parameters for a watershed model, and their approach requires data management, GIS processing, and relies upon proprietary software and scripting languages. These tools are valuable but require a high level of process knowledge, can require significant time investment, and can be difficult to apply in large, high resolution models. Bhatt et al. (2008) presented software for developing input data for the IHM called PIHM through integration with QGIS, providing input and output data processing tools (Qu and Duffy, 2007). Presently, automated software is not readily available that can be used to routinely and consistently generate input for GSFLOW; tools and methods that are available require complicated and informal workflow processing on diverse geospatial datasets. The tools being presented herein are intended to formalize the input data construction using a consistent approach, automate and simplify data manipulation and data transfer, support user intervention for customization, and provide robust data input that is suitable for IHMs.

This paper presents new software developed in the Python programming language that can be used to automatically generate required data input for GSFLOW. Although these scripts are tailored to create data input formats required for GSFLOW, the scripts could be adapted to develop input data for other IHMs that rely on regular grids for spatial discretization. For example, these scripts could be used to automatically develop a consistent grid-scale DEM and stream network appropriate for simulating surface water and groundwater interactions, and maps of climate, land cover, and land use distributed to each model grid cell. A stream network must be congruent with the model grid used for groundwater simulation for robust and convergent calculation of hydraulic gradients between streams and groundwater. Deriving model input for large-scale IHMs that produce convergent solutions remains a challenging and time-consuming process. The software presented herein was developed to overcome these challenges and to provide an efficient, reproducible, and automated procedure for GSFLOW input data construction. In the past, IHM development has required extensive GIS and computer programming expertise, which has restricted the use of IHMs in water resources management. This toolkit, called Gsflow-Arcpy, significantly simplifies GSFLOW model development, thereby potentially expanding the benefits of this IHM to a broader user community.

Gsflow-Arcpy automates many of the complicated and time-consuming GIS procedures necessary for parameterizing IHMs, including generating input representing the stream network, land cover, and meteorology distributions over complex terrains, such as steep and varying topography or flat regions where it can be difficult to establish the locations of stream channels. Stream network development based on digital elevation models (DEMs) are needed to determine the paths of water, sediment, and contamination movement (Tarboton, 1997). Similar to TOPMODEL (Beven and Kirkby, 1979) and HYDROTEL (Fortin et al., 2001), Gsflow-Arcpy uses a directional matrix and computes a topographic index to automatically determine drainage pathways. Unlike these previous studies that were solely focused on surface drainage, the present work extends to the simulation of surface water and groundwater exchanges, using equal area model grid cells for the groundwater model, and correctly overlaying the streams onto the groundwater model cells. Streams that are incorrectly placed onto grid cells will cause numerical problems. Specifically, the model grid-scale DEM and stream network are made consistent with respect to relative altitudes between cells containing streams, cells adjacent to streams, and the slope of cells following the streams. This prevents inconsistencies between the model grid DEM and the stream network that can lead to excessive surface water and groundwater exchanges, excessive spring discharge rates, and poor IHM convergence.

Gsflow-Arcpy provides data input requirements for the groundwater model component of an IHM, including the stream network and topology, and the top altitude for model cells that are used to define altitudes for aquifer boundaries. These groundwater parameters are developed at the same resolution as the model top grid. Additionally, this toolkit provides the necessary data sets for creating input for the Streamflow Routing Package, which has been one of the greatest challenges for developing MODFLOW and GSFLOW models. Additional processing outside of Gsflow-Arcpy is required to generate the hydrogeologic framework used to characterize aquifer systems and the associated MODFLOW input data. There are many separate software tools available that can be used for the development of input data that characterize aquifer systems. MODFLOW has benefited from many options for automated model construction that has led to broad application around the world (e.g., Leapfrog Hydro, 2013; Winston, 2009; Bakker et al., 2016). These existing tools are readily available and can be used in concert with Gsflow-Arcpy to provide capabilities to characterize aquifer systems using approaches more appropriately done outside the GIS environment.

2. Background

GSFLOW is an IHM developed by the USGS that integrates the watershed runoff model PRMS with the groundwater-flow model MODFLOW-NWT (Markstrom et al., 2008, 2015; Niswonger et al., 2011). GSFLOW was developed for application to medium and large-scale watersheds, (i.e., 10s–1000s of square kilometers), and has primarily been used as a tool for water resources management. GSFLOW has been applied to many basins around the world (see for example, Gannet et al., 2017, Hassan et al., 2014, and Wu et al., 2014, for examples from the US, Spain, and China, respectively), and each model application has generally relied upon varying approaches for model input construction. The difficulty in implementing required model input datasets for GSFLOW and other IHMs has led to recent work to develop a consistent set of tools for model input construction, primarily following the work of Huntington and Niswonger (2012), Niswonger et al. (2014), and Rajagopal et al. (2015).

Input data required for GSFLOW and other IHMs are derived from diverse geospatial datasets available for the continental United States and other places across the globe. These datasets are typically available through centralized data portals and can be downloaded freely to local computers for selected basins of interest. Gsflow-Arcpy was not designed to access geospatial datasets through web portals because these portals are not static and changes in data formats and storage locations make automated communication with web portals unreliable. Geospatial datasets must be downloaded and stored locally. Users may incorporate other datasets if available, so long as the data are in an acceptable format (i.e., raster files); users can refer to the formats for the standard data sources to develop datasets that rely on non-standard data sources. Gsflow-Arcpy develops the model input by reading geospatial datasets and making calculations using these data for each

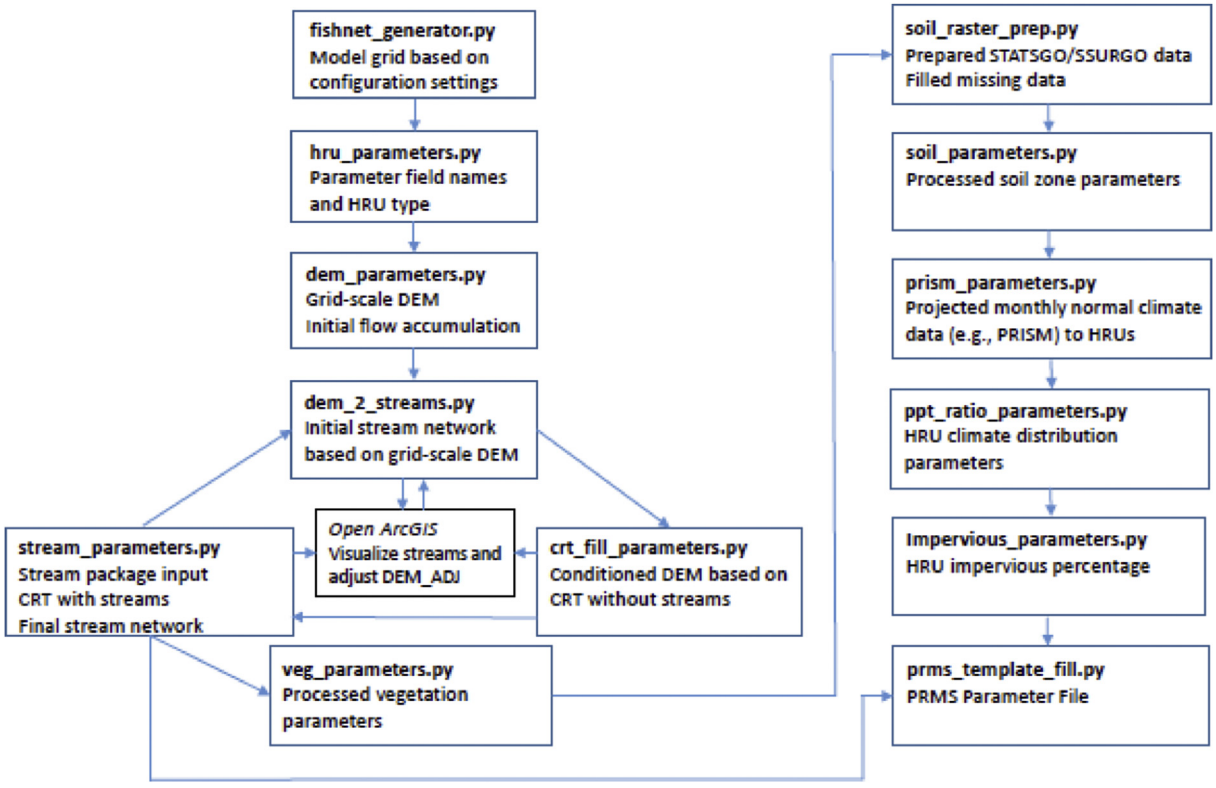

Fig. 1. Diagram illustrating the process flow for Gsflow-Arcpy and the 13 Python scripts that carry out the automated procedure for developing GSFLOW input datasets. The Cascade Routing Tool (CRT; Henson et al., 2013) is used by Gsflow-Arcpy.

PRMS parameter and MODFLOW dataset according to methods described by Viger and Leavesley (2007), Markstrom et al. (2008), Markstrom et al. (2015), Harbaugh (2005), Niswonger et al. (2011), and Henson et al. (2013). Most of the data required by Gsflow-Arcpy can be downloaded for basins in the United States using the USDA Geospatial Data Gateway: https://gdg.sc.egov.usda.gov/. However, some datasets must be obtained through regionally specific data sources, such as local agencies that collect and maintain these datasets. Examples of regionally specific data are climate and streamflow data that often are collected and managed locally, and these data must be obtained by regionally specific data collection efforts. Additionally, other regionally specific data include driller's logs or geophysical datasets used to develop the hydrogeologic framework for the groundwater system.

The process of extracting the surface drainage network from a DEM that is appropriate for IHMs remains a difficult process in complex terrain, despite well documented approaches (Bhatt et al., 2008; Peckham, 1998; Maidment, 2002; Daniels et al., 2011; Niu et al., 2014). Stream networks that are derived using previous approaches can potentially contain streams that float above and adjacent to stream canyons, are overly incised below the model grid cell top altitude, traverse uphill in the downstream direction, or form loops with ambiguous flow directions. Previous approaches have relied on high resolution DEMs to generate stream networks, as opposed to the method used by Gsflow-Arcpy that applies a DEM that has been up-scaled to the model grid resolution. Furthermore, no consideration is provided for constructed channels and canals that cannot be generated using topographic approaches for stream generation. Inconsistencies between a stream network and the model grid-scale DEM are problematic for IHMs that explicitly route surface water and calculate exchanges between surface water and groundwater and spring discharge using hydraulic gradients dependent on the relative altitude between streams, land surface, and groundwater head (Prudic et al., 2004; Niswonger and Prudic, 2005; Markstrom et al., 2008). Other errors in stream networks include dead ends (common issue where ephemeral stream channels become obscured where they flow across alluvial plains), misalignment relative to USGS National Hydrography Dataset (NHD) lines or aerial photos, and incorrectly located confluences. Stream network issues also arise when streams are derived from fine scale DEMs in arid regions where stream channels are undefined, or for DEMs that have not been properly conditioned to large grid-scales in complex terrain/drainage networks. DEM conditioning that can solve many of these problems include filling local swales that are artifacts of DEM construction, assuring that the grid cells have smooth and continuous downward slopes in the downstream direction, confirming that grid cells that contain streams have lower altitudes relative to adjacent grid cells, and creating and connecting intermittent stream reaches where channels are not well defined by topography. In some cases, dealing with these issues requires some level of subjectivity and the Gsflow-Arcpy scripts provide the option for user intervention to support these special cases.

A model grid-scale DEM and surface water network are generated by Gsflow-Arcpy using an iterative algorithm that conditions the model grid-scale DEM by filling local swales, creating smoothly downward sloping streamlines, and assuring that streams are not floating above or overly incised below model cell tops. This approach uses the drainage networks provided by NHD as a guide; however, the final directional matrix and topographic index provided by Gsflow-Arcpy is generated using model grid-scale DEM and flow accumulation and flow direction functions (Band, 1986; Maidment, 2002). Deriving the stream network from the model grid-scale DEM provides consistency of scale between streams and model grid cells that is appropriate for IHMs; however, depending on the resolution of the model grid, the location and altitude of streams may deviate from reality when using coarse grids.

3. Python scripts and the data development process

Developing a GSFLOW model using Gsflow-Arcpy can be organized into a set of distinct steps. Input data are generated for both the PRMS

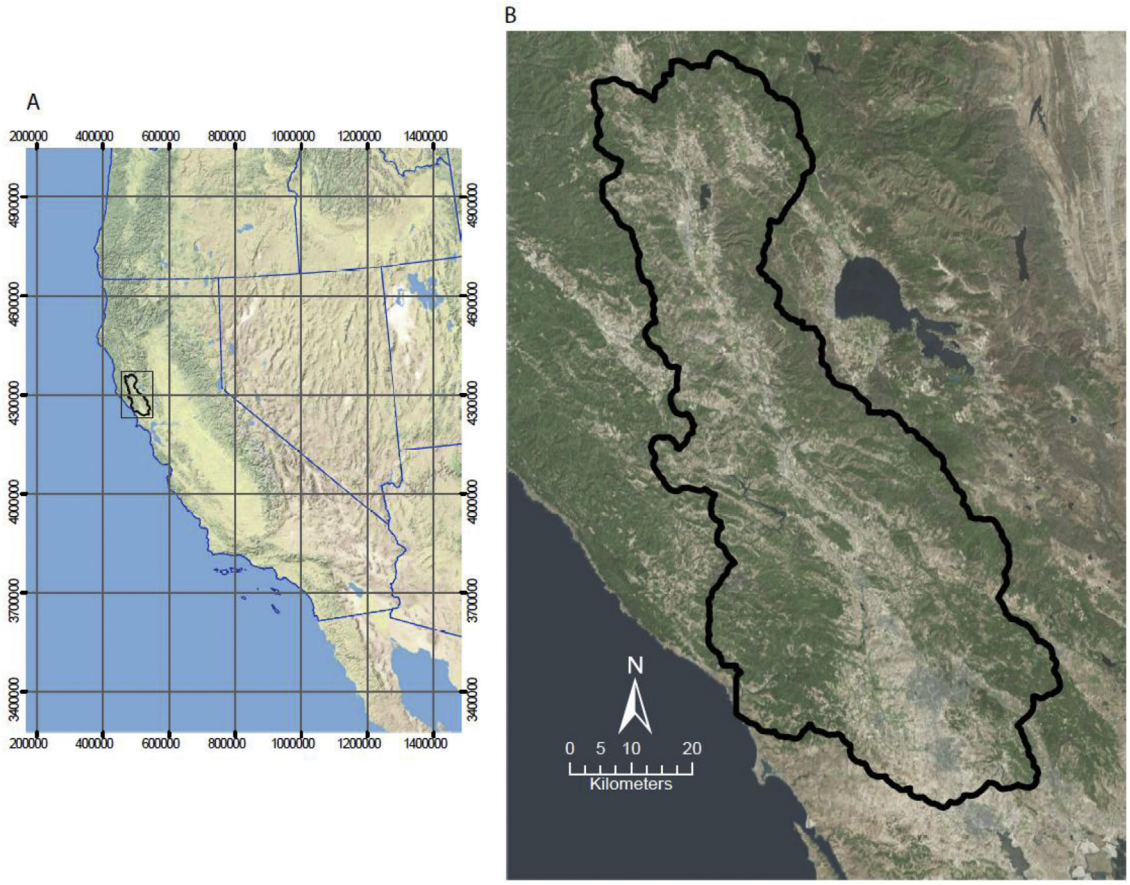

Fig. 2. Location of the Russian River watershed in northern California.

and MODFLOW components of GSFLOW. The procedure is carried out by 13 separate scripts that handle various workflow procedures (Fig. 1). These procedures include: data retrieval and preparation; model boundary delineation and horizontal discretization; grid-scale DEM initialization; grid-scale DEM fill and conditioning and surface water network delineation; lateral overland and shallow subsurface flow cascade routing delineation; land use, vegetation, and soil zone parameterization; climate distribution parameterization; and model input file construction. Each of these procedures represents a series of processes or commands that are carried out by running one or more Python scripts that automate spatial data processing required for GSFLOW model input.

Each script within Gsflow-Arcpy reads input files from a local folder and performs the operations necessary to project, transform, calculate, and format the data into an attribute table that is used to build the PRMS and MODFLOW inputs. Along with the scripts, a template Configuration File, field list file, remap files, template PRMS control file, and two CSV files that hold dimension and parameter settings are provided. The Configuration File guides the scripts to the correct folder locations, and the CSV files contain site specific settings. The field list holds the names of each field (column) that will be created in the shapefile. The remap files list unique IDs to transfer (remap) vegetation or soil codes to PRMS values. As the scripts are run, folders and files appear in the working directory, designated as the folder that contains the Configuration File. The parameter shapefile contains cell attributes and can be examined, edited, and displayed in ArcMap after running each script to visualize and evaluate intermediate results.

The Configuration File is a text file that acts as a guide for Gsflow-Arcpy. The purpose of the Configuration File is to avoid having to change the Python scripts for different models or model implementations. Each section of the Configuration File applies to a certain script and provides file locations and model specific settings and flags. The location of this file is considered the working directory that contains all folders and files created by the scripts. Several flags can be turned on or off in the Configuration File, depending on preferences of the user and project specifics. The field list file, field_list.ini, comes with the package and contains a list of the parameter names that make up the attribute table of the shapefile generated by Gsflow-Arcpy.

Gsflow-Arcpy discretizes PRMS and/or GSFLOW model areas using equal area square grid cells. For Gsflow-Arcpy models, PRMS hydrologic response units (HRUs) are coincident with MODFLOW grid cells. Thus, a single horizontal spatial discretization is used for both PRMS and all MODFLOW layers. Spatial units are called HRUs in PRMS, and grid cells in MODFLOW. However, Gsflow-Arcpy refers to both PRMS and MODFLOW spatial units as HRUs when these spatial units are equal. GSFLOW allows for different spatial discretization between PRMS and MODFLOW, and a user could create HRUs that are larger or finer than the MODFLOW grid cells with these scripts; however, this would require additional data handling outside the standard application of Gsflow-Arcpy to first create gridded HRUs and then separately create the MODFLOW grid and stream network. For this case, additional parameters would be required by GSFLOW that are not created by Gsflow-Arcpy to provide the connectivity between HRUs (Markstrom et al., 2008). Previous testing of GSFLOW models indicate that maintaining congruent HRUs and grid cells provides several advantages. These advantages include ease of integration between the surface water and groundwater systems, promoting model convergence and consistency by maintaining exchanges in water between storage reservoirs of similar volume, and the ability to represent lateral convergent and divergent cascading flows according to variations in altitudes among gridded HRUs. Gsflow-Arcpy uses a Cascade Routing Tool (CRT; Henson et al., 2013) to define lateral cascade pathways from HRU to

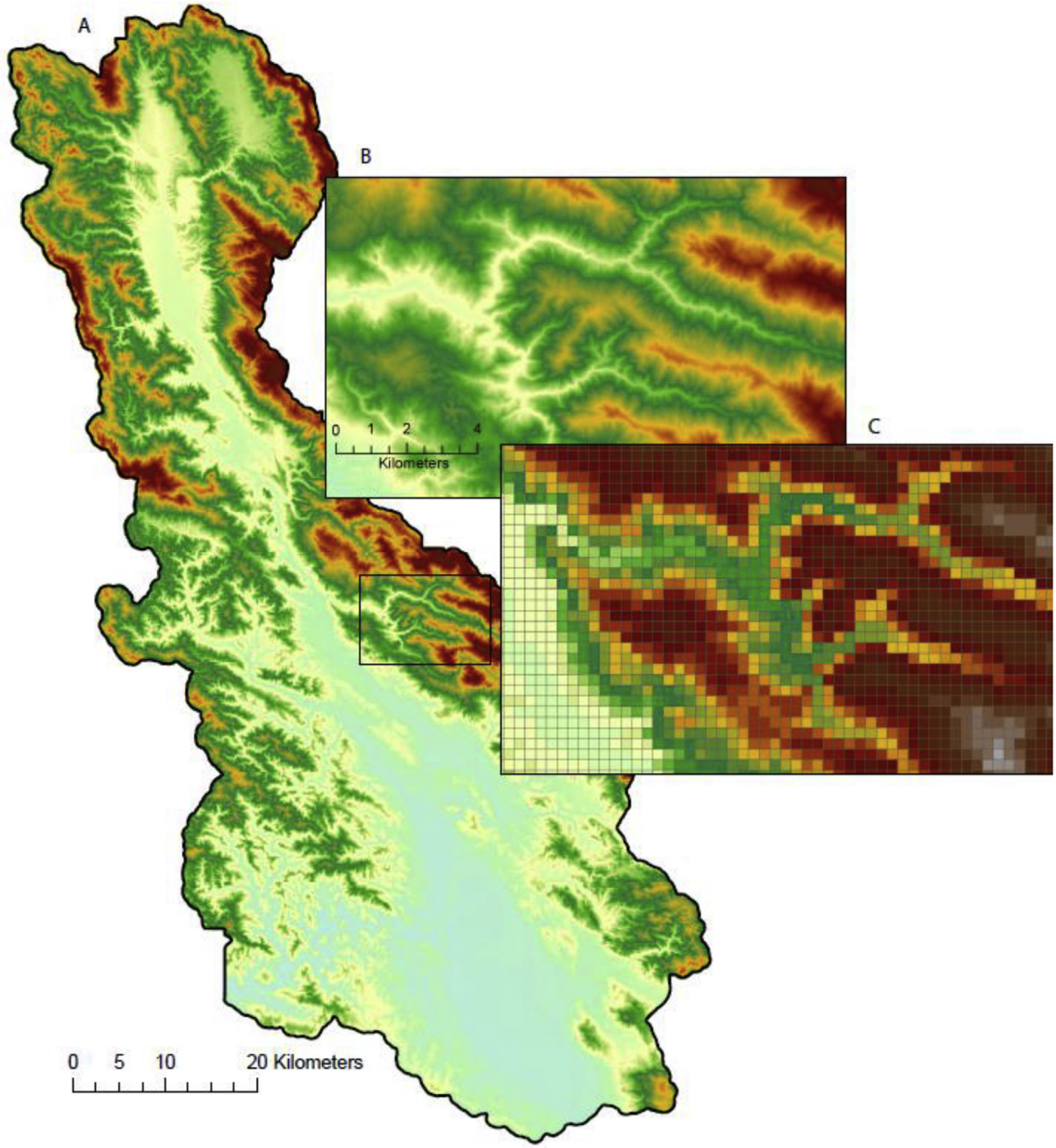

Fig. 3. Maps showing: A) Original 10 m DEM, clipped to the Russian River watershed, B) inset at the original DEM scale, and C) inset of the 300 m DEM re-sampled for the model grid.

HRU and HRU to streams and lakes. The CRT provides reach and segment numbering and defines the cascades based on the model grid-scale DEM.

4. Example application

Gsflow-Arcpy is presented and demonstrated herein by applying the tools to develop the input datasets for a GSFLOW model of the Russian River watershed located in northwestern California (Fig. 2). The Russian River traverses 177 km north to south through Sonoma and Mendocino counties and drains an area of approximately 3850 km^2. The Russian River watershed provides a useful example application as it has complex geography (over 600 m of hilly relief alongside very low-relief floodplains) and a range of small streams that feed the main stem of the Russian River. Permeable aquifers and stream sediments result in a high degree of connectivity between streams and groundwater, and GSFLOW provides a useful model for evaluating water resources in the basin (Woolfenden and Nishikawa, 2014).

5. Geospatial and other ancillary data requirements

For the case of the Russian River watershed, the DEM and the Russian River drainage area (Fig. 2) were downloaded from the USDA Geospatial Data Gateway. The model boundary follows the watershed drainage divide for the Russian River watershed. For cases where the model boundary does not align with true watershed boundaries, custom polygon shapefiles can be derived from a DEM using ArcGIS. Any polygon dataset that defines a land based study area can be used with Gsflow-Arcpy.

Spatially distributed vegetation type and percent cover was derived

from the LANDFIRE dataset (http://www.landfire.gov/vegetation.php), and spatially distributed soil data were derived by STATSGO and SSURGO datasets (also available through the USDA Geospatial Gateway). Impervious area data were obtained from the National Land Cover Database (NLCD http://www.mrlc.gov/nlcd11_data.php). Climate datasets used to spatially distribute daily precipitation and maximum and minimum temperatures are typically derived by combining daily climate measured at stations located in the model boundary with spatially distributed monthly climatologies of these variables. Monthly climatologies (i.e., 30-year monthly averages) were obtained from the Parameter Elevation Regression on Independent Slopes Model (PRISM) climate portal (http://www.prism.oregonstate.edu/normals/). Daily climate measured at climate stations located in the Russian River watershed were attained from National Oceanic and Atmospheric Association (NOAA) coop stations (http://www.wrcc.dri.edu/climatedata/climsum/) and from local agencies. A stream network for the Russian River was attained from NHD (https://nhd.usgs.gov/NHD_High_Resolution.html). The NHD stream network was used as a guide for generating the model grid-scale stream network used in the GSFLOW model.

6. Gsflow-arcpy workflow

The workflow for generating GSFLOW input follows the sequential application of each Python script while displaying output in ArcGIS to evaluate the results. Scripts must be run in the order shown below, with exceptions noted, as results from one script are required to run subsequent scripts. However, in some cases, such as for developing a stream network, a subset of scripts can be run iteratively to correct and improve results to better represent the system. The individual scripts are described below in order of their execution.

6.1. fishnet_generator.py

The fishnet generator script uses ArcPy functions to discretize the model domain into square HRUs and to spatially reference the model grid. The term fishnet is used to describe the generated model grid in ArcGIS upon which the model parameters are developed. This also is referred to as the parameter shapefile because it is a shapefile in ArcMap that stores parameters for each HRU. For the Russian River model, HRUs were set to 300 × 300 m (Fig. 3b). This script reads cell size and buffer cells from the Configuration File and uses ArcPy tools to create the fishnet grid shapefile, based off the study area boundary extent and spatial projection. Alternatively, an existing model grid shapefile can be used for Gsflow-Arcpy by setting the spatial location of the lower left corner of the grid and skipping to hru_parameters.py.

6.2. hru_parameters.py

The HRU parameters script constructs the fields for the attribute table of the model parameter shapefile, based on those provided in the field_list.ini Configuration File. The attribute table will display each of these field names as column headers. These columns start out blank, and fill with data as the scripts are run. This script also will designate the HRU type as inactive, land, lake, or swale based on the study area boundary and shapefiles designating lakes and model points. The lake shapefile, if applicable, is typically extracted from an NHD dataset, or can be manually created.

6.3. dem_parameters.py

The DEM parameters script populates the elevation fields by resampling the DEM to the model grid resolution (Fig. 3). DEM parameters are used to develop the elevation dataset, stream network, and cascade parameters for GSFLOW. Minimum, maximum, and mean DEM values are all written to the parameter shapefile, providing the user with several options for model grid elevations. This script also runs a flow accumulation function to weight stream elevations over the surrounding elevations to better represent streams as the DEM values are scaled to the fishnet. This helps with stream network development and cascade routing and numbering.

6.4. dem_2_streams.py

The DEM to streams script uses the HRU elevations generated in dem_parameters.py to build the stream network, applying the topographical index of the model-grid scale DEM for the directional matrix. This is the first of the three steps in the iterative process of stream network development. The initial flow accumulation from dem_parameters.py helps ensure that streams will be in the correct location. This script runs the Arc Hydro flow accumulation and flow direction tools, using threshold settings from the Configuration File, and to build the initial stream network (Maidment, 2002). The flow accumulation threshold designates the minimum number of upslope drainage HRUs required to generate a stream that will be explicitly represented in the model. The flow length threshold sets a minimum length for a potential stream reach; streams shorter than this threshold are excluded. Adjusting these values can give the user a range of stream network detail (Fig. 4). Water flowing in streams is routed as channelized flow, whereas lateral surface flow on HRUs that do not contain a stream is routed as sheet flow (Markstrom et al., 2008). Results from this step can be evaluated in ArcMap to examine the stream network. Even with the flow accumulation function, stream deviations can occur when the DEM is scaled to the model-grid scale, so some adjustments may be necessary to better match streams to NHD flowlines and/or aerial photos. Cell altitudes can be manually adjusted in ArcGIS by modifying the DEM_ADJ values in the HRU parameter shapefile attribute table (Fig. 5). These adjustments are purely up to the user, and not necessary to continue. The CRT (in the following script) will provide several elevation adjustments to fill swales that will allow continuous flow of the stream network and cascades. Areas where manual adjustments are commonly applied are floodplains, confluences, and lake inlets/outlets.

6.5. crt_fill_parameters.py

The CRT fill parameters script runs the Cascade Routing Tool (Henson et al., 2013) without streams. This script identifies each HRU that has no drainage pathway to any of the 8 adjacent HRUs for 8-way flow direction, referred to as a swale or sink. CRT will not fill swales that contain a stream, so this step runs CRT without streams to ensure smooth and continuous downward slopes along streams. The amount an HRU elevation must be filled to create at least one drainage pathway to another HRU or model outflow point is recorded in the parameter shapefile. The user can turn on a flag to automatically apply the fill amounts to DEM_ADJ values and continue, which will eliminate (most, if not all) swales within two iterations. However, the user may choose to manually change the DEM_ADJ values (as shown above), or use a combination of CRT fill values and manual adjustments, such as in areas where larger fills are required. Fig. 6 provides an example of how Gsflow-Arcpy is used to create clean flow paths in complex areas where NHD streams are inconsistent with HRU elevations.

6.6. stream_parameters.py

After swales are filled, the stream parameters script runs the CRT with streams to generate the final cascade routing parameters for GSFLOW (Fig. 7). Cascade parameters define the direction and proportion of overland and shallow subsurface flow routing used to calculate lateral flows from HRU to HRU, and HRU to streams, lakes, and model outflow points. The script also generates the Streamflow Routing Package (SFR2; Niswonger and Prudic, 2005) input values for MODFLOW. These data are all added to the parameter shapefile attribute

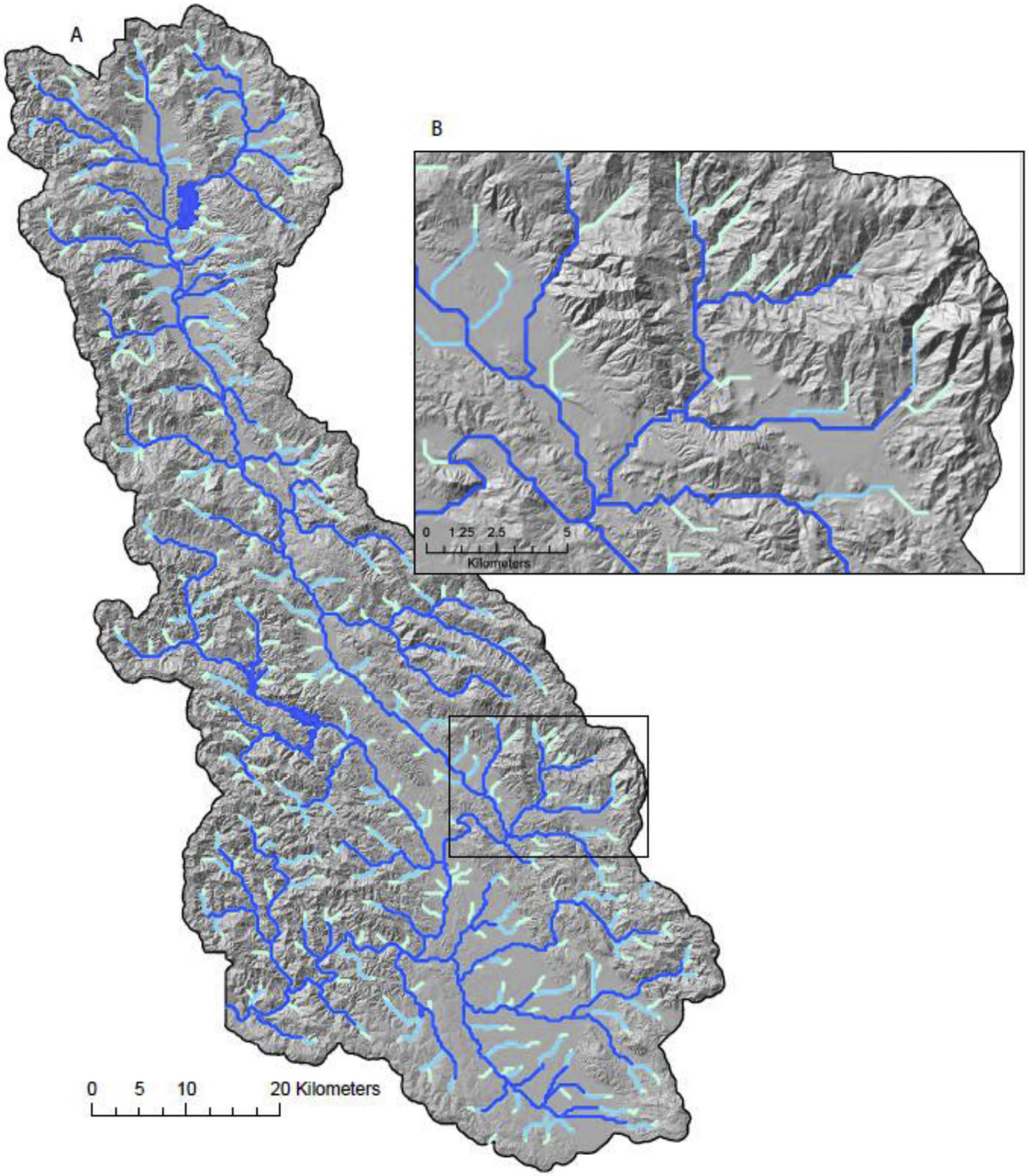

Fig. 4. Stream networks generated with different flow accumulation and flow length threshold settings; A) Russian River watershed, and B) inset of a sub-basin showing different stream resolutions. The darkest blue lines (streams_100_10) show a network with a flow accumulation threshold of 100 and flow length threshold of 10. HRUs with flow accumulation and flow lengths smaller than the specified threshold will not contain streams. The lighter blue lines show the levels of detail that can be obtained by decreasing the flow accumulation and stream length thresholds (streams_50_5 and streams 30_3, respectively). These thresholds are site specific and depend on the resolution and purpose of the model. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

table. This is the final step in the iterative stream network development process. All stream network scripts read DEM_ADJ for HRU elevation; additional changes made to DEM_ADJ at this point would require that the stream scripts be re-run, starting from dem_2_streams.py.

6.7. *veg_parameters.py*

The vegetation parameters script generates all PRMS parameters derived from the LANDFIRE vegetation dataset, including coverage type, winter and summer coverage density, winter and summer precipitation canopy interception, and root depth (Fig. 8). The script applies remap files to transfer LANDFIRE vegetation values (EVT120, EVT130, or EVT140) to corresponding PRMS values. For example, a LANDFIRE value of 3015 (CA Coastal Redwood Forest), would be remapped to a PRMS vegetation type of 4 (conifer). These remap files are provided with the scripts, and can be modified to include additional vegetation types for a region. The script produces vegetation maps with PRMS values that are used to populate the parameter shapefile attribute

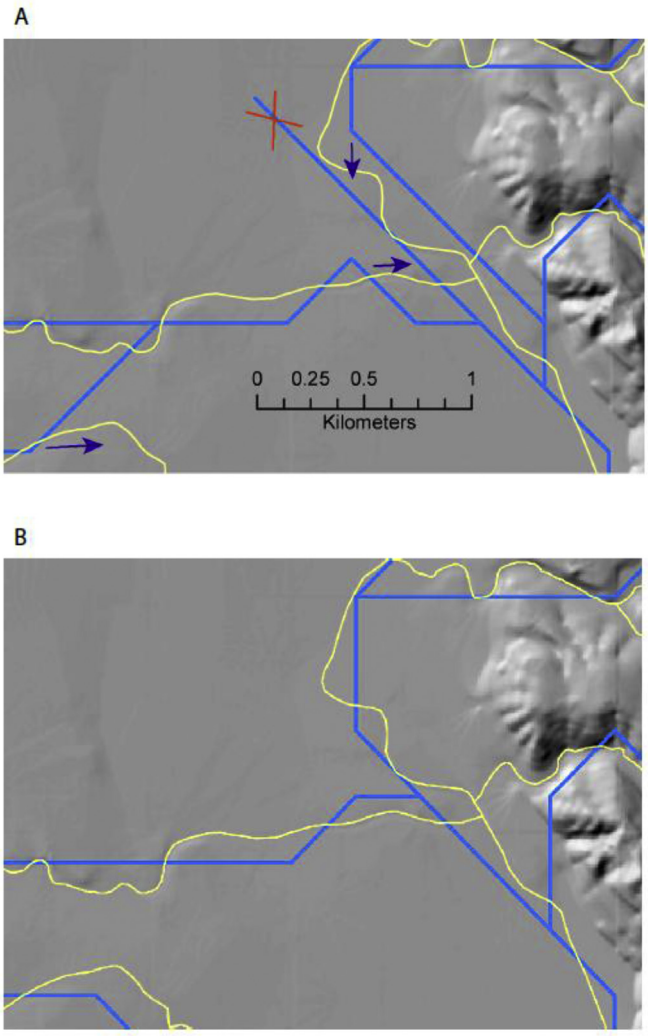

Fig. 5. Gsflow-Arcpy provides the user with the option to make manual adjustments to the DEM_ADJ values in the HRU parameter shapefile attribute table to guide the streams in certain areas, such as flat regions or drainages that are very close to each other, potentially causing a segment to 'jump' drainages: A) an erroneous segment (marked with a red X), and the directions (blue arrows) of the desired stream paths; and B) the resulting flow paths after re-running dem_2_streams.py with the adjusted elevations. The amount of manual adjustments is completely up to the user, as some may choose to use the CRT fill values and move on, while others may want to fine tune their stream network beyond the capability of the scripts. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

table. This script must be run before the soil scripts such that vegetation dependent root depth is available to calculate root zone soil parameters.

6.8. soil_raster_prep.py

The soil preparation script projects the input soil data rasters to the spatial projection of the fishnet and clips each raster to the extent of the model domain. Input rasters include percent of sand, silt, and clay, available water capacity, and saturated hydraulic conductivity. Soil depth can be optionally provided, and soil zone calculations will use the greater of root depth vs. soil depth. These data are provided with the SSURGO/STATSGO data, and must be imported into ArcGIS, using the Soil Data Viewer tool, where they can be converted to rasters. If gaps exist in the soil data, the script will operate a data-fill procedure using adjacent (nearest neighbor) data values. It is recommended that STATSGO data are used to cover any areas where SSURGO data are unavailable before using adjacent data values. Soil_raster_prep.py must be completed before running soil_parameters.py.

6.9. soil_parameters.py

The soil parameter script uses the prepared soil rasters of percent sand, silt, and clay, along with available water capacity and saturated hydraulic conductivity to derive GSFLOW soil zone parameters (Fig. 9). Soil depth is derived from the greater of root depth and soil depth (if provided) from STATSGO or SSURGO and is incorporated into the attribute table by veg_parameters.py. These initial parameterizations are often modified during calibration by scaling the parameters equally for all cells located in each gaged subbasin. A flag can be set in the Configuration File to read a user-supplied surficial geology shapefile for deriving soil parameters that control deep percolation beneath the soil zone or shallow aquifer hydraulic conductivity. The geology will be re-mapped into user determined classifications based on a remap table, for example; consolidated rock, unconsolidated rock, and basin fill. These general geologies will hold values that will be applied to the soil parameter derivations. This allows for more realistic initial values of certain soil parameters that affect the transfer of water through the subsurface.

6.10. prism_parameters.py

Gsflow-Arcpy uses PRISM or other spatial climatology datasets to distribute climate station data to all HRUs in a manner that incorporates the effects of slope, aspect, and altitude. Previous applications of this script have relied upon the PRISM 30-year climatologies for the period 1981–2010 at an 800 m resolution. The PRISM parameters script determines the PRMS parameters that are used to distribute daily maximum and minimum temperatures and a precipitation value for each HRU during the GSFLOW model runtime. This distribution is

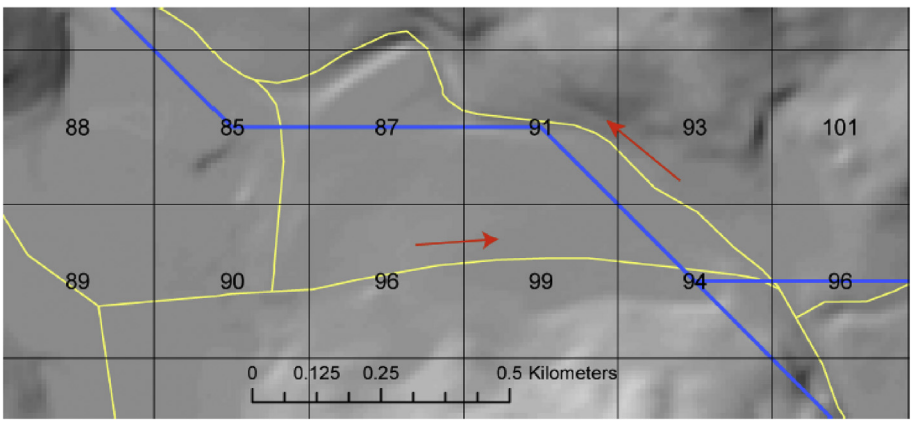

Fig. 6. Inconsistencies between the USGS National Hydrographic Database (NHD) streamlines and the resampled grid-scale DEM based streams are reconciled using Gsflow-Arcpy and the Cascade Routing Tool (CRT). This example shows NHD lines that end up having uphill flow paths across HRUs with higher elevation resulting from up-scaling the DEM to the model grid resolution. Yellow lines are NHD flowlines and red arrows indicate original flow direction. The blue lines are the stream lines created by Gsflow-Arcpy. Numbers are model grid top altitudes (rounded for viewing), and black lines are model grid lines. This is an example of why NHD flowlines cannot be directly applied to the model. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

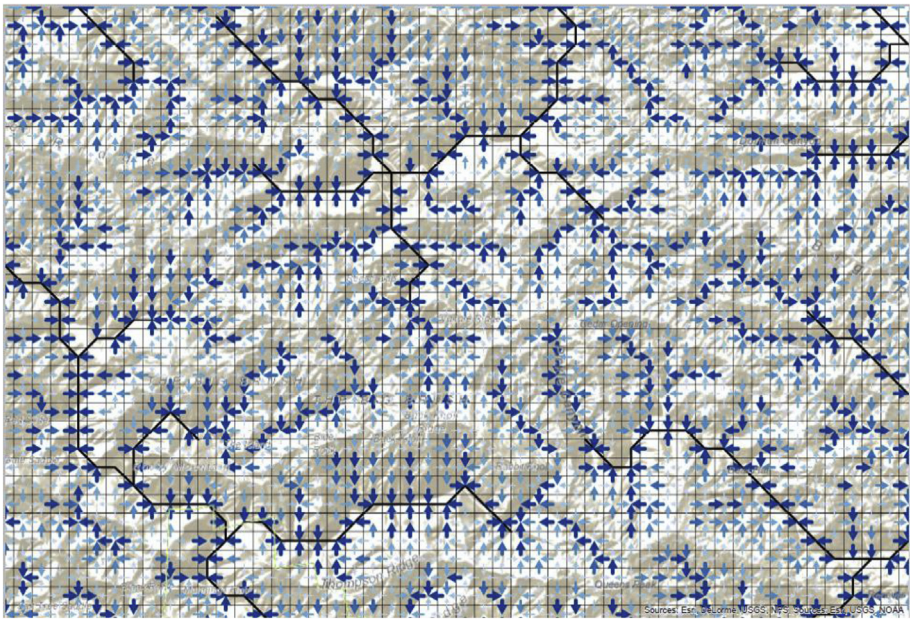

Fig. 7. Inset from Russian River model showing HRUs, streams, and cascading flow directions created by the Cascade Routing Tool (CRT). Cascade parameters generated by CRT are used for distributed overland flow and shallow subsurface flow routing in GSFLOW (Henson et al., 2013). Darker colored arrows indicate greater drainage area accumulation.

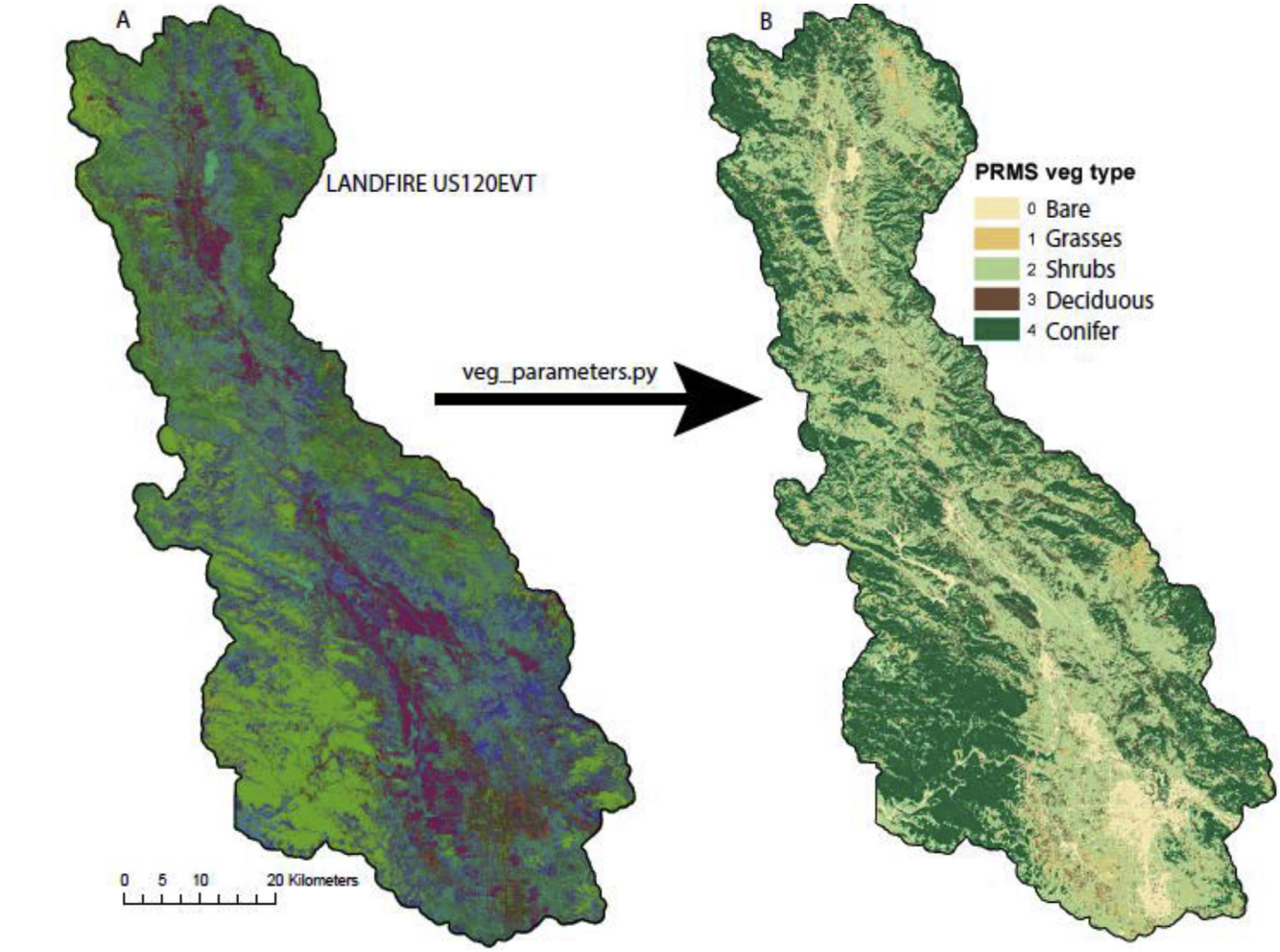

Fig. 8. Map showing vegetation type remapped from: A) LANDFIRE existing vegetation type (US120EVT), and to B) PRMS values. Vegetation type is just one of the several vegetation based parameters that veg_parameters.py converts to PRMS values for each HRU.

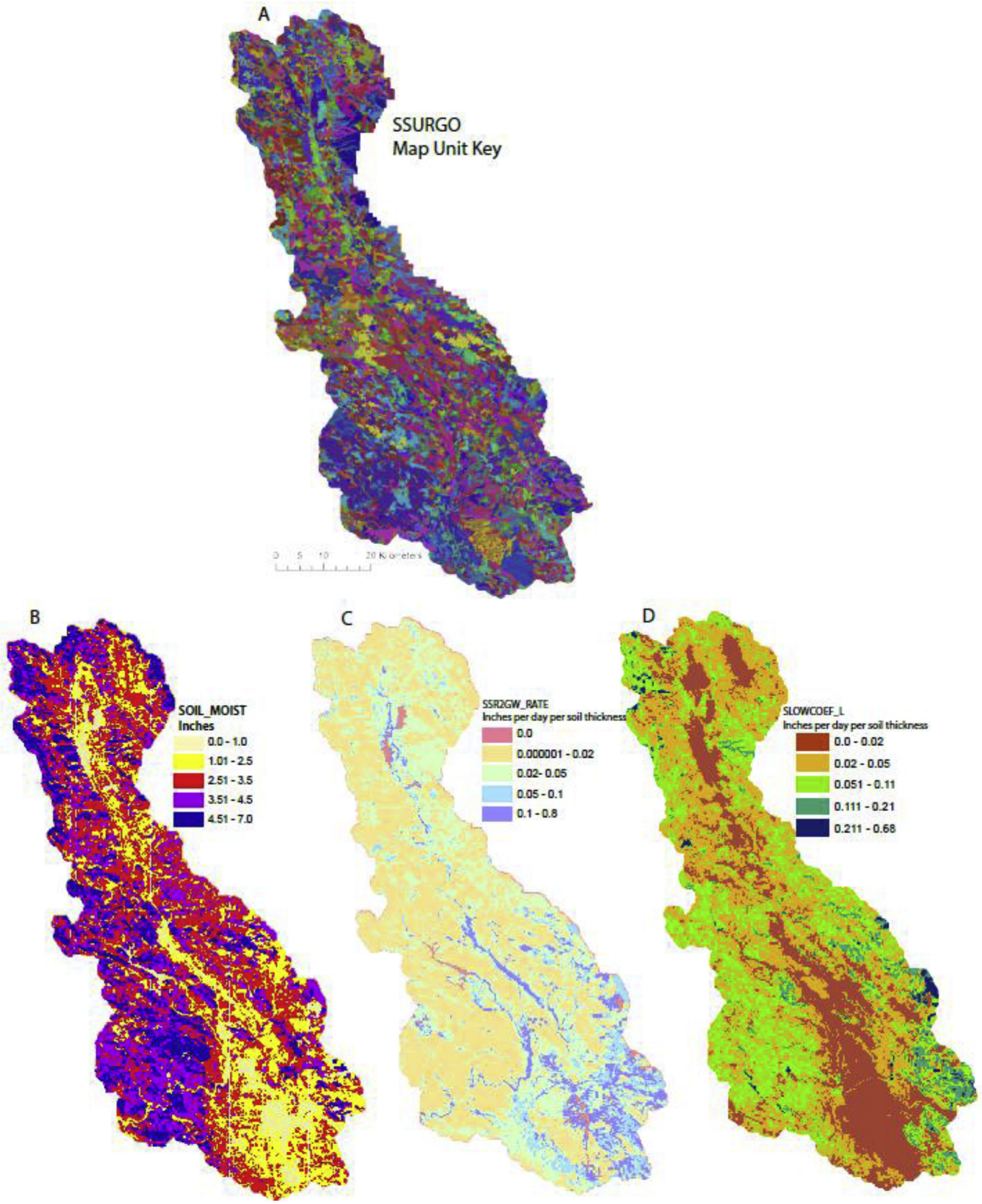

Fig. 9. Maps showing: A) the mosaic of the SSURGO map unit keys for the Russian River watershed and resulting rasters of soil properties; B) MOIST_MAX (PRMS parameter soil_moist_max); C) SSR2GW_RATE (PRMS parameter ssr2gw_rate); and D) SLOWCOEF_L (PRMS parameter slow_coef_lin).

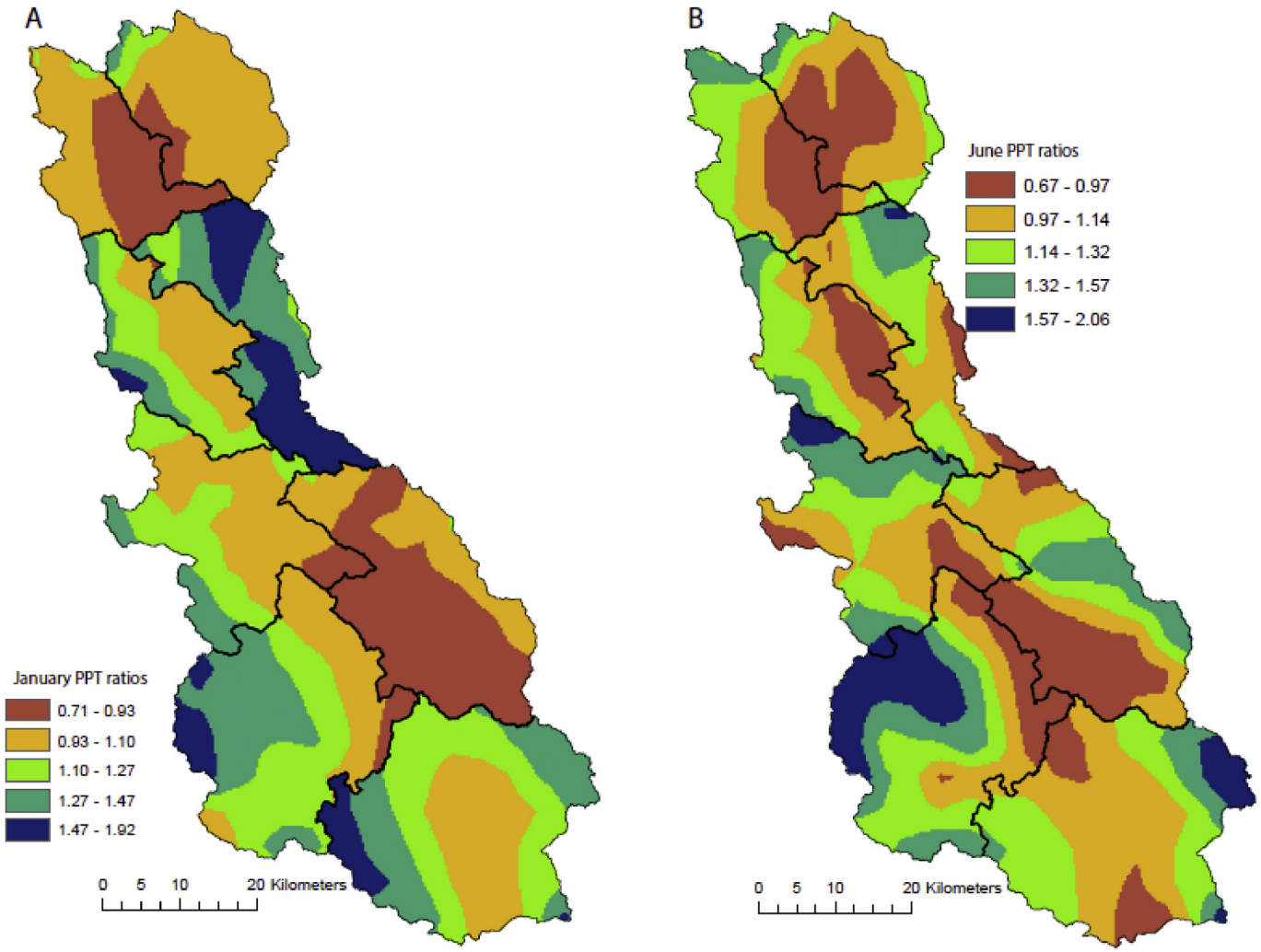

Fig. 10. Precipitation factors for A) January, and B) June calculated from monthly average PRISM values and daily precipitation values measured at climate stations located within the watershed. Nine precipitation stations and respective zones were used in this example. Precipitation factors are calculated for each month.

based on the relationship between observed station data and the gridded climate dataset provided. The factors derived from this relationship is used to extrapolate climate station data over the model domain. These GSFLOW climate parameters can be derived from different gridded resolutions and/or time periods (i.e., DAYMET), depending on the model simulation period or other project requirements, but the script is tailored to read PRISM data without any reformatting. The script operates ArcMap's zonal statistics tool to resample the PRISM gridded data to HRUs, along with projecting, transforming, clipping, and snapping the data to HRUs. This script must be run before ppt_ratio_parameters.py.

6.11. *ppt_ratio_parameters.py*

The precipitation ratio script calculates precipitation factors assigned to each HRU and each month. These factors are calculated as the ratio between PRISM data and respective observed mean monthly precipitation values from a designated climate station or stations. These spatially distributed monthly factors for each HRU are then multiplied by the climate station(s) values to distribute daily precipitation to all HRUs for each day in the simulation period (Fig. 10). A flag is set in the Configuration File to read the climate station identification number for each HRU. Precipitation zones can be provided by the user to apply multiple climate stations to the precipitation ratio calculations. The precipitation ratios are written to the shapefile attribute table as monthly ratios per HRU, and ultimately written to the PRMS Parameter File as the rain/snow adjustment factors (Markstrom et al., 2008).

6.12. *impervious.py*

The impervious script uses impervious cover data from the National Land Cover Database to derive the percentage of impervious surface within each HRU. The script also generates the impervious percentage for all HRUs.

6.13. *prms_template_fill.py*

The template fill script writes the PRMS Parameter Files, either in column or array format (based on settings in the configurations file). A single Parameter File can be written if older versions of PRMS are being used. Required PRMS dimensions that are read by this script, and written to the PRMS Parameter Files must be specified by the user in CSV files that accompany the scripts. These CSV files hold basic settings, such as specific model dimensions and required default values for parameters not defined by the ancillary data described above. Dimensions and default parameter values can be adjusted according to the needs of a model application. Example templates of the CSV files are provided with Gsflow-Arcpy. A subset of the input data required for MODFLOW, including the input for the Streamflow Routing Package and model top altitude required for the Discretization Package can be

exported from the attributes of the parameter shapefile.

7. Supplementary files

Another Python file called support_functions.py is included in Gsflow-Arcpy. This script contains functions that are used by other scripts within Gsflow-Arcpy. This file must be present in the directory with the rest of the Python scripts, but does not need to be run by the user. Currently, Gsflow-Arcpy does not create the data and control files that are required for a GSFLOW simulation. Templates for these files that can be used for manually creating these files are included with the Gsflow-Arcpy toolkit. Although the MODFLOW grid, Streamflow Routing Package data input, and optionally layer thicknesses can be created by Gsflow-Arcpy, the suite of MODFLOW input files that include these data are not generated by Gsflow-Arcpy. These MODFLOW input files can be created using other software products that are currently available, such as ModelMuse (Winston, 2009) or Flopy (Bakker et al., 2016) in the public domain or one of the several commercial software packages.

Data sets for running these scripts on the Russian River are available by request through the corresponding authors. Because the data files for the Russian River watershed are very large, example data input for another example is included as supplementary material to this paper. Supplementary material for this paper includes the application of the Sagehen watershed in California, USA (Markstrom et al., 2008). The Russian River example was presented here as it symbolizes a large, complex model domain that suits the capabilities of GSFLOW to simulate diverse model conditions. The Sagehen examples provided with the toolkit are small model builds meant to be used by both the developers and users as quick, single issue test models.

8. Discussion

Development of Gsflow-Arcpy was motivated by the need to automate a very arduous and time-consuming process required to convert diverse geospatial datasets into IHM input. Another important aspect of this work is the development of an approach that is transferable to other IHM modeling environments for creating models with a stream network topology that is consistent with the model grid-scale DEM. Here we address additional complexities in generating a stream network required for simulating surface water and groundwater exchanges, whereas previous work on methods for generating stream networks in watersheds strictly considered surface drainage processes. This process is challenging, especially when using large HRUs to discretize watersheds with steep topography. Furthermore, a robust and consistent stream network development process is important for constructing stable and efficient IHMs. Structural model errors due to poorly constructed geometries and parameters used in coupled governing equations for surface water and groundwater can severely hinder IHM applications. Specifically, the algorithms in Gsflow-Arcpy for constructing stream networks and elevations help ensure robust simulation of groundwater and surface water exchanges.

Gsflow-Arcpy combines automation with the ability for user intervention and manual manipulation of data sets to ensure flexible model input data construction. User intervention is facilitated by interfacing data construction with ArcGIS to provide visualization of intermediate products, and specialization of data manipulation when required. Gsflow-Arcpy was developed recognizing the need to create large scale models that require resampling of DEMs to large HRUs. This process inevitably creates artifacts in the model grid-scale altitude that must be fixed with some sacrifice in DEM accuracy (Wilson, 2012). However, benefits from enhanced model efficiency attained through coarse discretization is often a necessity that must justify losses in resolution. Important future work is the evaluation of optimal IHM resolution that will no doubt benefit from automated and reproducible model input construction, as provided by Gsflow-Arcpy.

Although beyond the scope of this present work, post processing and visualization of model results is another important component of any study using an IHM for hydrologic investigation. Because GSFLOW simulates all major hydrologic processes in watersheds, large amounts of output are produced making calculation of water budgets and visualization more difficult and important. Existing software for MODFLOW (e.g., ModelMuse and Flopy) can be used to visualize output from GSFLOW; however, software that can be used to analyze and visualize GSFLOW results from a wholistic perspective is limited. These authors hope that with the development of software for creating data input for GSFLOW, more applications will lead to additional work to develop post processing software for GSFLOW.

9. Conclusions

Integrated modeling is increasingly used to support water resources management and for understanding natural and anthropogenic impacts on natural resources. A new Python toolkit called Gsflow-Arcpy presented herein is intended to formalize much of the model input data construction using a consistent approach. This paper introduces and demonstrates its capabilities to provide a streamlined, reproducible process for creating input data required for the integrated hydrologic model GSFLOW. A fully constructed PRMS Parameter File is created by the package, along with several MODFLOW input parameters. This approach automates data manipulation, formatting, transfer, and GIS processing, and provides robust data input suitable for IHMs. Although these scripts are tailored to fulfill input requirements for GSFLOW, the datasets produced are general, and these scripts could be adapted to develop input data for any IHM that relies on regular grids for spatial discretization. The current version of Gsflow-Arcpy does not convert the groundwater inputs into model ready input, as it does for PRMS input files. The process can be combined with other automated approaches, such as Flopy, to fully complete the GSFLOW model input construction. Gsflow-Arcpy is demonstrated through application to the Russian River watershed, a 3850 km^2 coastal watershed located in northwest California, USA. The Russian River watershed provides a useful demonstration of the toolkit's capabilities because it contains complex topography and a highly dendritic stream network that typically make it challenging to model surface water and groundwater interactions. Gsflow-Arcpy significantly simplifies the model development process which may enhance accessibility of integrated hydrologic models to a broader user group that can benefit from decision support tools for managing natural resources.

Acknowledgments

Research supported by grant from the Water Sustainability and Climate Program jointly funded by the National Science Foundation (1360506) and U.S. Department of Agriculture/National Institute of Food and Agriculture (1360507). Support was also provided by the U.S. Geological Survey's Water Use and Availability Program. We would like to thank Paul Barlow with the U.S. Geological Survey, and three anonymous reviewers, for their timely and insightful colleague reviews. All data used in the analysis and to support our conclusions in this manuscript may be obtained from MAG (e-mail: mgardner@usgs.gov) or RGN (e-mail: rniswon@usgs.gov). Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.envsoft.2018.07.020.

References

Bakker, M., Post, V., Langevin, C.D., Hughes, J.D., White, J.T., Starn, J.J., Fienen, M.N., 2016. Scripting MODFLOW model development using Python and FloPy. Ground Water 54 (5), 733–739.

Band, L.E., 1986. Topographic partition of watersheds with digital elevation models. Water Resour. Res. 22 (1), 15–24.

Beven, K.J., Kirkby, M.J., 1979. A physically based, variable contributing area model of basin hydrology/Un modèle à base physique de zone d'appel variable de l'hydrologie du bassin versant. Hydrol. Sci. J. 24 (1), 43–69.

Bhatt, G., Kumar, M., Duffy, C.J., 2008. Bridging gap between geohydrologic data and integrated hydrologic model: PIHMgis. In: Paper Presented at iEMSs 2008. International Congress on Environmental Modelling and Software, Barcelona.

Clark, M.P., Nijssen, B., Lundquist, J.D., Kavetski, D., Rupp, D.E., Woods, R.A., Freer, J.E., Gutmann, E.D., Wood, A.W., Brekke, L.D., Arnold, J.R., 2015. A unified approach for process-based hydrologic modeling: 1. Modeling concept. Water Resour. Res. 51 (4), 2498–2514.

Daniels, M.H., Maxwell, R.M., Chow, F.K., 2011. An algorithm for flow direction enforcement using subgrid-scale stream location data. J. Hydrol. Eng. 16, 677. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000340.

Fatichi, S., Vivoni, E., Ogden, F., Ivanov, Z., Mirus, B., Gochis, D., Downer, C., Camporese, M., Davison, J., Ebel, B., Jonesk, N., Kim, J., Mascaro, G., Niswonger, R., Restrepoo, P., Rigon, R., Shen, C., Sulis, M., Tarboton, D., 2016. An overview of current applications, challenges, and future trends in distributed process-based models in hydrology. J. Hydrol. 537, 45–60.

Fortin, J.P., Turcotte, R., Massicotte, S., Moussa, R., Fitzback, J., Villeneuve, J.-P., 2001. A distributed watershed model compatible with remote sensing and GIS data. Part 1: description of the model. J. Hydrol. Eng. 6 (2), 91–99.

Gannett, M.W., Lite Jr., K.E., Risley, J.C., Pischel, E.M., La Marche, J.L., 2017. Simulation of Groundwater and Surface-Water Flow in the Upper Deschutes Basin, Oregon. U.S. Geological Survey Scientific Investigations Report 2017-5097, 68 p.. https://doi.org/10.3133/sir20175097.

Gruber, S., Peckham, S., 2009. Land-surface parameters and objects in hydrology. Dev. Soil Sci. 33, 171–194.

Harbaugh, A.W., 2005. MODFLOW-2005, the U.S. Geological survey modular groundwater model—the ground-water flow process: U.S. Geological Survey Techniques and Methods 1 6-A16, (various pagings).

Hassan, T.S.M., Lubczynski, M.W., Niswonger, R.G., Su, Z., 2014. Surface-groundwater interactions in hard rocks in Sardon Catchment of western Spain: an integrated modeling approach. J. Hydrol. 517, 390–410. https://doi.org/10.1016/j.jhydrol.2014.05.026.

Henson, W.R., Medina, R.L., Mayers, C.J., Niswonger, R.G., Regan, R.S., 2013. CRT—cascade Routing Tool to define and visualize flow paths for grid-based watershed models. U.S. Geological Survey Techniques and Methods 6-D2, 28.

Huntington, J.L., Niswonger, R.G., 2012. Role of surface-water and groundwater interactions on projected summertime streamflow in snow dominated regions: an integrated modeling approach. Water Resour. Res. 48 (11).

Hydro, Leapfrog, 2013. ARANZ Geo Limited. Available from: http://www.leapfrog3d.com.

Jasiewicz, J.Ł., Metz, M., 2011. A new GRASS GIS toolkit for Hortonian analysis of drainage networks. Comput. Geosci. 37 (8), 1162–1173.

Laniak, G.F., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N., Blind, M., Peckham, S., 2013. Integrated environmental modeling: a vision and roadmap for the future. Environ. Model. Software 39, 3–23.

Maidment, D.R., 2002. Arc Hydro: GIS for Water Resources, vol. 1 ESRI, Inc.

Markstrom, S.L., Niswonger, R.G., Regan, R.S., Prudic, D.E., Barlow, P.M., 2008. GSFLOW-coupled groundwater 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 Techniques and Methods 6-D1, 240.

Markstrom, S.L., Regan, R.S., Hay, L.E., Viger, R.J., Webb, R.M.T., Payn, R.A., LaFontaine, J.H., 2015. PRMS-IV, the precipitation-runoff modeling system, version 4: U.S. Geological Survey Techniques and Methods 158. book 6, chap. B7. https://doi.org/10.3133/tm6B7.

Metz, M., Mitasova, H., Harmon, R.S., 2011. Efficient extraction of drainage networks from massive, radar-based elevation models with least cost path search. Hydrol. Earth Syst. Sci. 15 (2), 667.

Ng, G.-H., Wickert, A.D., Somers, L.D., Saberi, L., Cronkite-Ratcliff, C., Niswonger, R.G., McKenzie, J.M., 2018. GSFLOW-GRASS v1.0.0: GIS-enabled Hydrologic Modeling of Coupled Groundwater-surface-water Systems. submitted for publication. Geoscientific Model Development.

Niswonger, R.G., Prudic, D.E., 2005. Documentation of the Streamflow-Routing (SFR2) Package to include unsaturated flow beneath streams - a modification to SFR1: U.S. Geological Survey Techniques and Methods 6-A13, 50.

Niswonger, R.G., Panday, S., Ibaraki, M., 2011. MODFLOW-NWT, A Newton formulation for MODFLOW-2005: U.S. Geological Survey Techniques and Methods 6-A37, 44.

Niswonger, R.G., Allander, K.K., Jeton, A.E., 2014. Collaborative modelling and integrated decision support system analysis of a developed terminal lake basin. J. Hydrol. 517, 521–537.

Niu, G.Y., Paniconi, C., Troch, P.A., Scott, R.L., Durcik, M., Zeng, X., Huxman, T., Goodrich, D.C., 2014. An integrated modelling framework of catchment-scale ecohydrological processes: 1. Model description and tests over an energy-limited watershed. Ecohydrology 7 (2), 427–439.

Peckham, S.D., 1998. Efficient extraction of river networks and hydrologic measurements from digital elevation data. In: Barndorff-Nielsen, O.E., Gupta, V.K., PérezAbreu, V., Waymire, E. (Eds.), Stochastic Methods in Hydrology: Rain, Landforms, and Floods. World Scientific, Singapore, pp. 173–203.

Prudic, D.E., Konikov, L.F., Banta, E.R., 2004. A New Streamflow-routing (SFR1) Package to Simulate Stream-aquifer Interaction with MODFLOW-2000. USGS Open File Report 2004-1042. pp. 104.

Qu, Y., Duffy, C.J., 2007. A semidiscrete finite volume formulation for multiprocess watershed simulation. Water Resour. Res. 43 (8).

Rajagopal, S., Huntington, J.L., Niswonger, R., Pohll, G., Gardner, M., Morton, C., Zhang, Y., Reeves, D.M., 2015. Integrated Surface and Groundwater Modeling of Martis Valley, California, for Assessment of Potential Climate Change Impacts on Basin-Scale Water Resources, vol. 4126. Desert Research Institute Publication, pp. 54.

Tarboton, D.G., 1997. A new method for the determination of flow directions and upslope areas in grid digital elevation models. Water Resour. Res. 33 (2), 309–319.

Tian, Y., Zheng, Y., Zheng, C., 2016. Development of a visualization tool for integrated surface water–groundwater modeling. Comput. Geosci. 86, 1–14.

Viger, R.J., Leavesley, G.H., 2007. The GIS Weasel User's Manual: U.S. Geological Survey Techniques and Methods, Book, vol. 6. pp. 201 chap. B4.

Wilson, J.P., 2012. Digital terrain modeling. Geomorphology 137 (1), 107–121.

Winston, R.B., 2009. ModelMuse-A graphical user interface for MODFLOW-2005 and PHAST: U.S. Geological Survey Techniques and Methods 6-A29, 52.

Woolfenden, L.R., Nishikawa, T., 2014. Simulation of groundwater and surface-water resources of the santa rosa plain watershed, sonoma county, California: U.S. Geological Survey Scientific Investigations Report 2014–5052 258. https://doi.org/10.3133/sir20145052.

Wu, Bin, Zheng, Yi, Tian, Yong, Wu, Xin, Yao, Yingying, Han, Feng, Liu, Jie, Zheng, Chunmiao, 2014. Systematic assessment of the uncertainty in integrated surface water-groundwater modeling based on the probabilistic collocation method. Water Resour. Res. 50 (7), 5848–5865.