PennState Integrated Hydrologic model (PIHM) Tutorial[1]

Contact: Mukesh Kumar (mukesh.kumar@ua.edu)

Version: A.0

Case: Second Creek Watershed

[1] Tutorial prepared by JuneHyeong Park and Mukesh Kumar in Feb. 2019, based on earlier versions. The model version used here is a result of research by several people, including Chris Duffy, Yizhong Qu, Mukesh Kumar, Gopal Bhatt, Xing Chen, Yanlan Liu, and Dongdong Wang.

Table of Contents[2]

---

[2] If you are interested in knowing the file format of PIHM input files, you may jump to section 9.

1 Introduction

PIHM (Penn State Integrated Hydrologic Model) is an open-source, physically-based, fully-distributed hydrologic model (Qu, 2005; Qu and Duffy, 2007; Kumar, 2009). The model uses unconstructed mesh for discretizing the model domain, which provides numerous advantages in terms of representational accuracy (Kumar et al., 2009) and computational efficiency (Wang et al., 2018). Running the model requires populating each structured mesh with meteorological and physiographic properties. This is done using PIHMgis (Bhatt, et al., 2014), an open-source, platform independent, extensible and tightly-coupled GIS interface to PIHM. PIHMgis uses a shared data model framework (Kumar et al., 2010) that allows quick generation and revision of the PIHM model input files.

The tutorial presents the entire PIHM model application workflow for a selected watershed. These steps can be repeated for any other watershed of your choice. The tutorial discusses how the users may obtain raw data, use PIHMgis to generate input files for PIHM using the raw data, and run the PIHM model.

2 Download PIHMgis

PIHMgis version 3.0 can be downloaded from the PIHMgis website (http://www.pihm.psu.edu/pihmgis_home.html) under **Downloads** tab (Fig. 2.1). You need to select the appropriate version according to your device's OS. Users are encouraged to use the Macintosh (Mac) version. Windows version has not kept up with the OS changes, and may lead to crashes. If you want to run PIHMgis in Linux system, you need to compile the source code natively and meet the library dependencies. You may download the PIHMgis source code from GitHub. This tutorial is based on the Mac version.

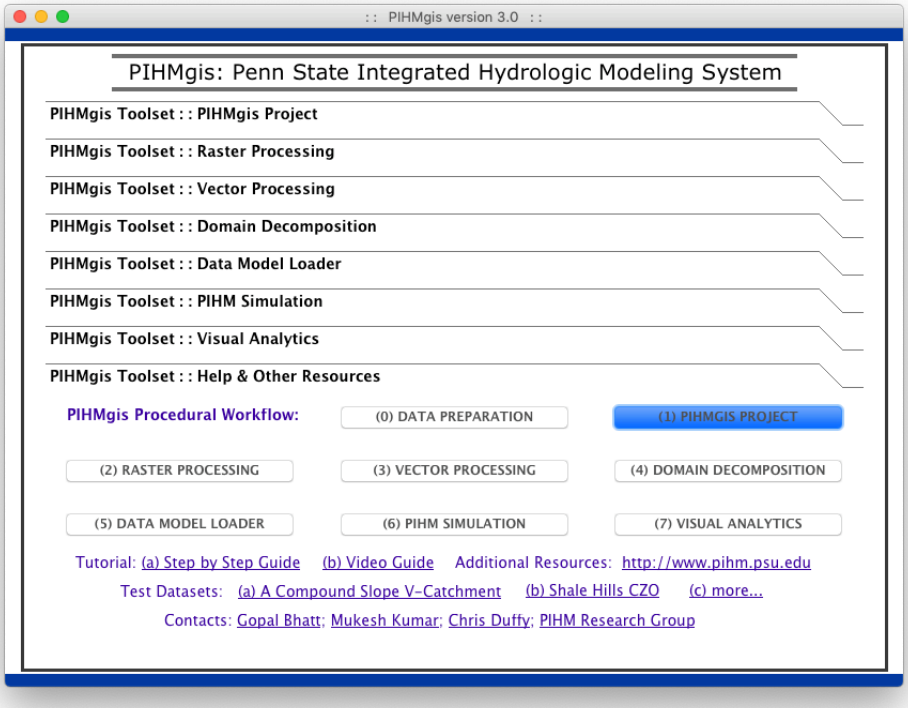


Figure 2.1 PIHMgis ver. 3.0

3 Download target watershed and related raw data

3.1 Target watershed

Identify the watershed location using How's my waterway (https://watersgeo.epa.gov/mywaterway/) or Science in your watershed (https://water.usgs.gov/wsc/map_index.html) tools, if your study area is in US. Otherwise use related tools. Now we need to download relevant data for modeling.

3.2 Data Requirement

Data needed for successfully setting up the model include:

Digital Elevation Model (DEM) and bedrock elevation map, Soil texture or soil hydrologic properties, Geology texture or hydrologic properties of the aquifer, Land cover and vegetation parameters, and Climate variables for the modelling period.

For this tutorial, for DEM, 'USGS NED 1 arc-second 2013 1 x 1 degree[3]' was used. For forcing data, 'NLDAS Primary Forcing Data L4 hourly

[3] TNM Download (https://viewer.nationalmap.gov/

0.125 x 0.125 degree V002[4]' was used. For land use and vegetation, 'NLCD 2011 Land Cover (2011 Edition, amended 2014)[3]' and 'Monthly vegetation parameters based on NLDAS land cover classes[5]' was used. For soil properties, 'SSURGO (Soil Survey Geographic Database)[6, 7]' was used. With these spatially distributed data set, you can run this model with heterogeneous input layers. Footnotes provide the link to locations for downloading these data for US watersheds.

4 PIHMgis Project

PIHMgis workflow is stored as a 'project'. Users can keep track of their input and output, and come back to a closed/saved project later and finish their task. There are four options under PIHMgis Project menu (Fig. 4.1).

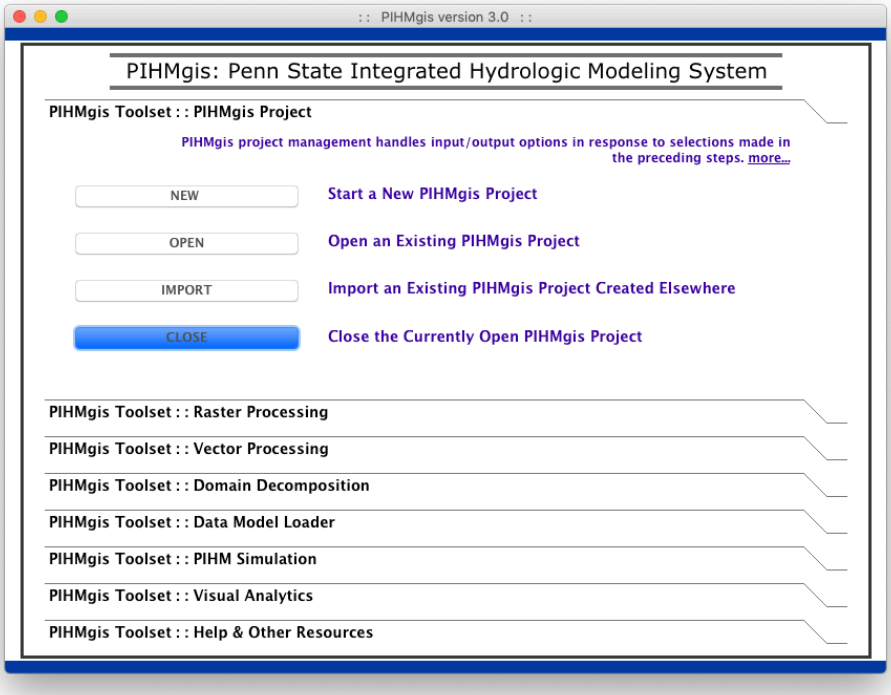


Figure 4.1 Four options under Project menu of PIHMgis

[4] NASA GES DISC (https://disc.gsfc.nasa.gov/datasets/NLDAS_FORA0125_H_V002/summary)

[5] NLDAS Vegetation Parameters (https://ldas.gsfc.nasa.gov/nldas/NLDASmapveg.php)

[6] Web Soil Survey (https://websoilsurvey.sc.egov.usda.gov/App/WebSoilSurvey.aspx)

[7] Soil Data Viewer (https://www.nrcs.usda.gov/wps/portal/nrcs/detailfull/soils/survey/geo/?cid=nrcs142p2_053620)

4.1 New (PIHMgis) Project

Your activities for creating PIHM input files using PIHMgis are all saved in the folder of each PIHMgis project. Fig. 4.2 shows the dialog box. When you press the 'Browse…' button, you can select a folder. When you select the folder, 'New (PIHMgis) Project module' creates a new PIHMgis-project-folder and some other sub-folders under the selected folder to organize all the intermediate outputs from individual PIHMgis modules. '*.pihmgis' file is also created within the project folder, and users can reopen an existing PIHMgis project with this file. As you can see in Fig. 4.2, project 'Nov2120181505' was created under the selected folder '~/ihlab/Documents/PIHM_JP/PIHMgis'. The name of project is automatically using the 'month-date-year-hour-minute' format.

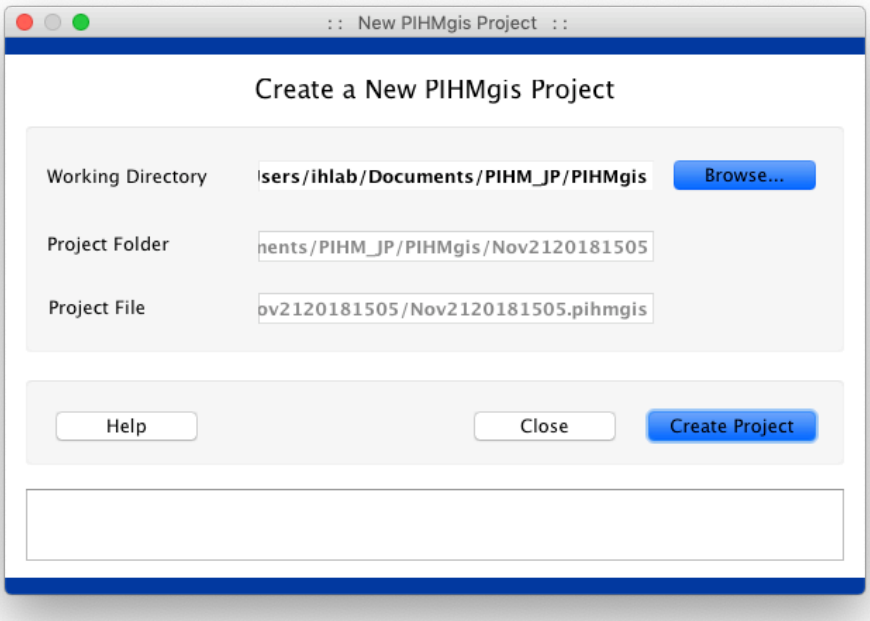


Figure 4.2 Create a new PIHMgis project dialog

4.2 Open Project

You can open any existing project, which was closed earlier or is not currently active. Note that if you closed a project but changed the location of the project folder, you need to import it first. As shown in Fig. 4.3, you may press the 'Browse…' button to open a project. After you make your selection and press the 'Open Project' button, it will load the project. Opening project is really important that when you proceeded to any specific individual module with the project, your

activities were recorded. So, you can see which modules were already used, which files were inserted as input, and which files were created.

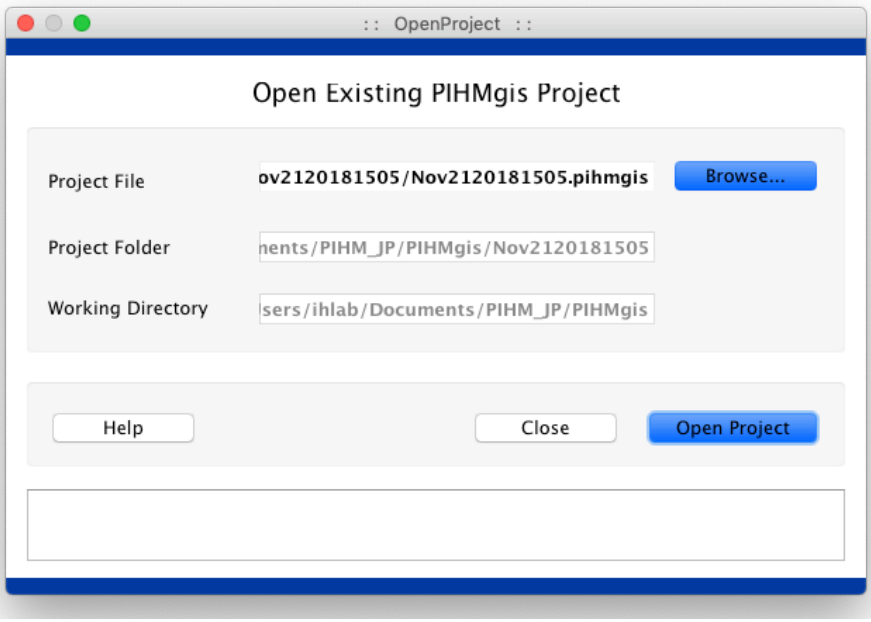


Figure 4.3 Open an existing PIHMgis dialog

4.3 Import Project

Import project module allows you to import an existing project. Since the path of the file is recorded in the PIHMgis project file is an absolute path – if you change the name of any directory on the root directory tree those files won't be accessible. Also, if you happen to move the folders around (including sending or receiving from different computers), Import project module will update the existing project file so that you can continue to work on the project.

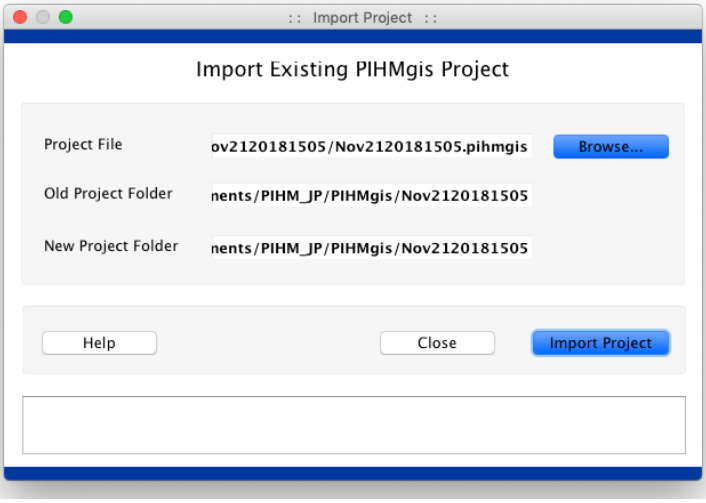


Figure 4. Import any existing PIHMgis project

4.4 Close Project

This option is useful when you work on multiple projects simultaneously. When you want to switch the project, close the current project with this option, and you can open the latter project. As shown in Fig. 4.5, you can terminate the current project with pressing 'Close Project' button. You can use this option to know the name of the currently opened project. This dialog shows you the current project.

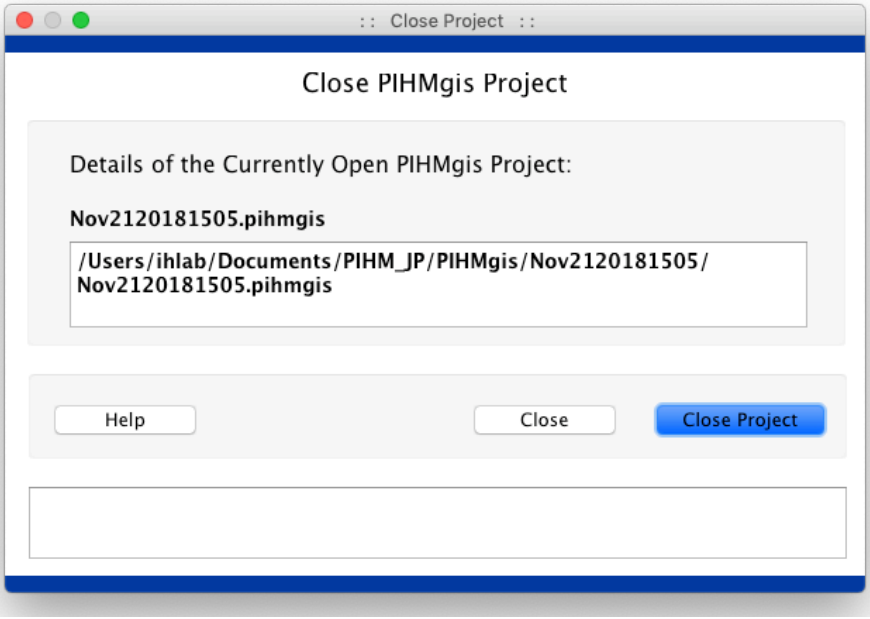


Figure 4.2 Close currently opened PIHMgis project

5 Raster Processing

5.1 Introduction

'Raster Processing' extracts subshed/watershed delineation and streams from Digital Elevation Model (DEM). These steps could be performed in most other GIS software (e.g., ArcGIS, QGIS etc.). Please use them if PIHM doesn't work as expected. Although, PIHMgis will create files exactly suitable for PIHM.

As shown in Fig. 5.1, you can download DEM from the national map viewer. This tutorial used 1-arc second resolution DEM.

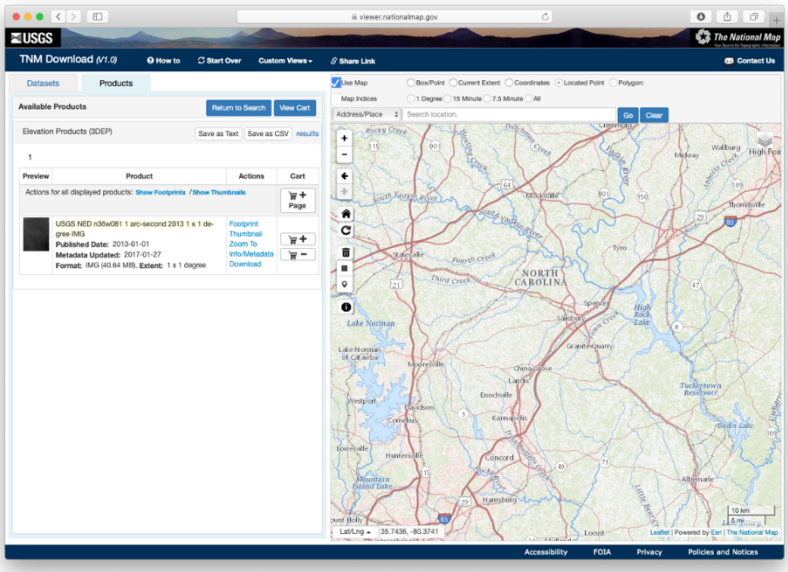


Figure 5.1 TNM Download website for DEM (http://viewer.nationalmap.gov)

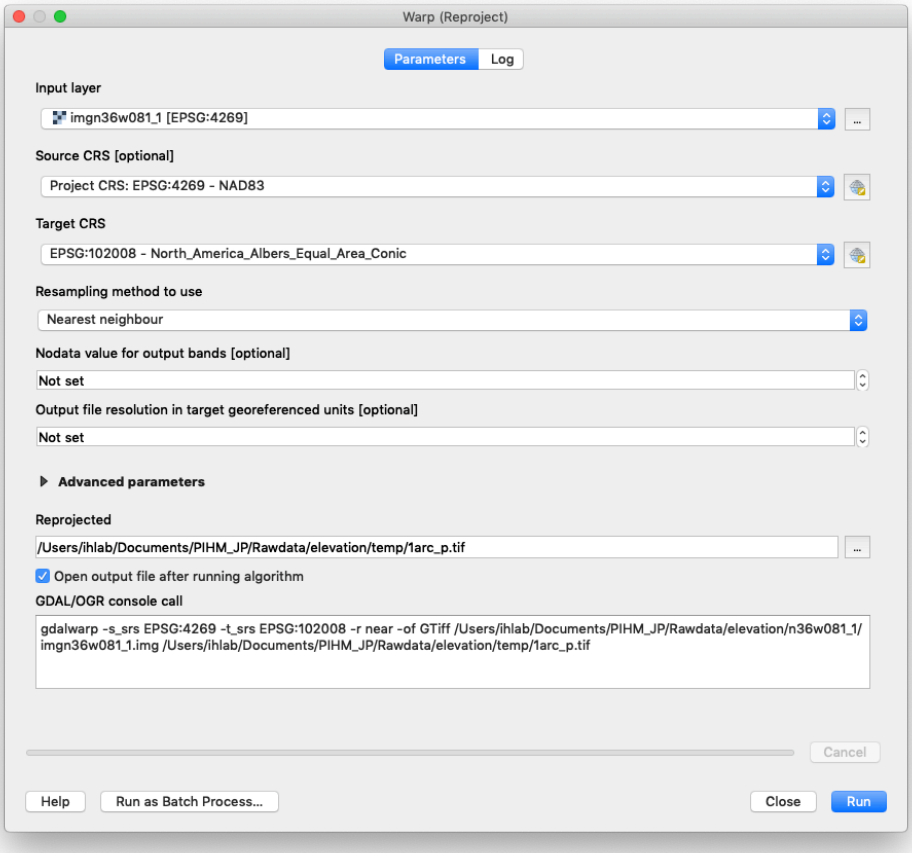


Figure 5.2 Change projection of DEM

After downloading the DEM, you may want to perform three more steps: change projection (Fig. 5.2), mask (Fig. 5.3), and change file format to ascii (Fig. 5.4). Changing the projection to one of the "projected coordinate systems" that uses Cartesian coordinates is essential. Mask operation is optional and should be used if you already have the shapefile for the watershed. In that case, first create a buffered shapefile for the watershed (buffer distance used should be ~2 times the DEM resolution). The mask option will produce a clipped DEM of the watershed. The alternative is to use the rectangular DEM, derive all watersheds within the rectangular boundary using the Raster processing operations, and then select out the watershed boundary of your interest. Finally, you need to save file format of DEM as 'ascii (.asc)' as shown in Fig. 5.4. Fig. 5.5 is the result of pre-processing on DEM before performing 'Raster Processing' operations.

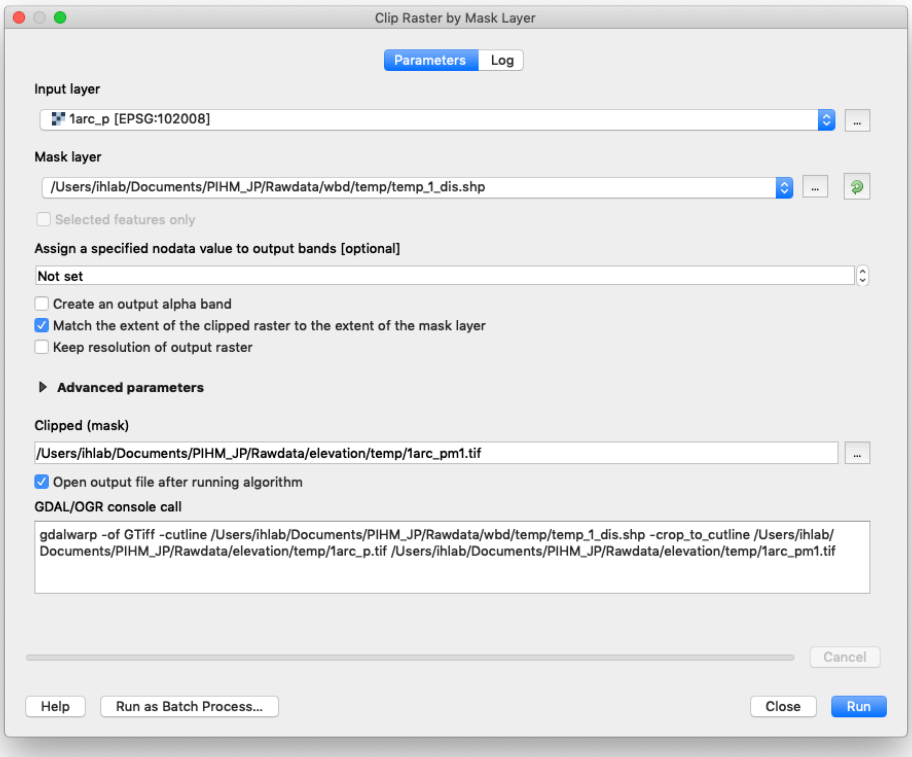


Figure 5.3 Mask DEM with dissolved target watershed shapefile

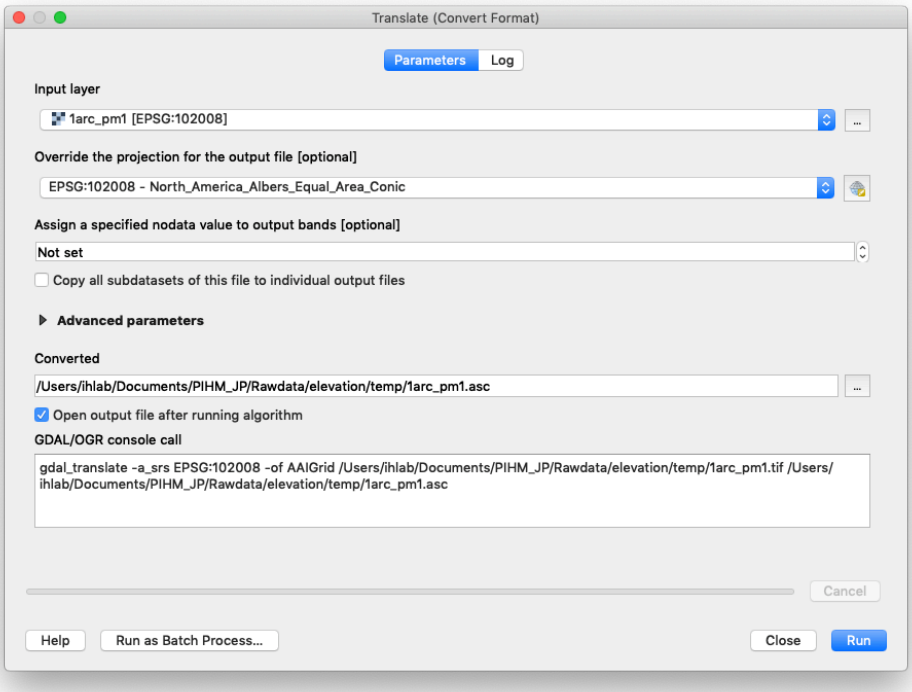


Figure 5.4 Change file format from other raster formats to ascii (.asc)

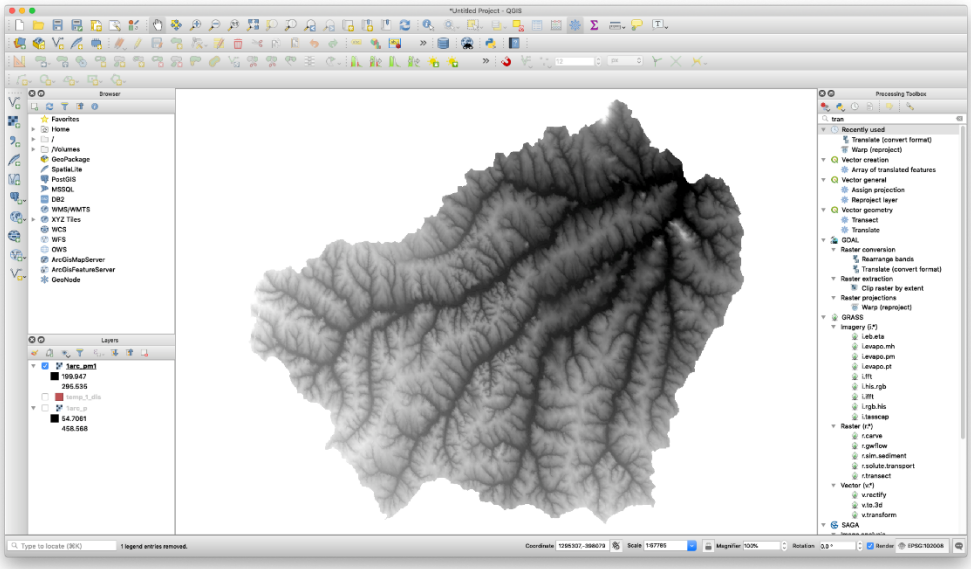


Figure 5.5 Results of pre-processing on DEM

5.2 Fill Pits

'Fill Pits' fills pits in each grid of the DEM. If a grid is surrounded by grids that have higher elevation, water should be trapped in that grid and cannot flow out. They are generally considered as artifacts that interfere with the routing of flow across grids, and are removed by raising their

elevation to the point where they essentially drain off the edge of the grid. In order to perform customized fill pit operations, users are referred to ArcHydro or TauDEM>

This module considers two raster formats, ESRI binary (.adf) and ascii (.asc). With pre-processing in chapter 5.1, you may have DEM file in ascii format. The Fill Pits processing could take several minutes depending on the size and resolution of DEM. The file format of output is also ascii. The text browser below provides information related to the status of processing. If the box of 'Load in GIS' in output dialog is checked, the output file will be opened automatically within your default GIS platform. If you want quick process without checking the output file, you can make that box empty as you can see in Fig. 5.6.

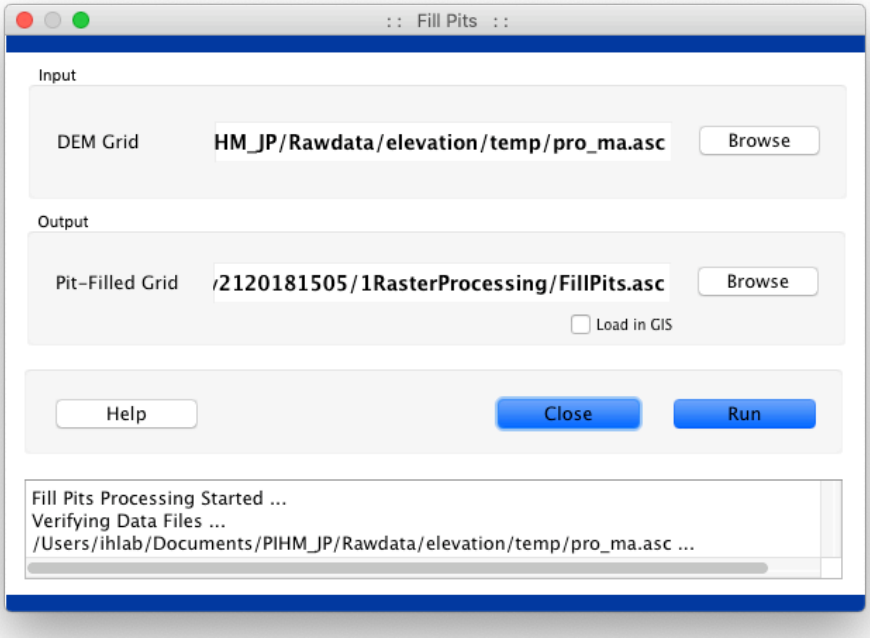


Figure 5.6 Fill Pits dialog

5.3 Flow grids

Using the output file of 'Fill Pits' module, this module calculates flow direction and flow accumulation. Flow direction is calculated based on the D8 algorithm (O'Callaghan and Mark, 1984) that the value of each grid indicates flow direction of each grid. Flow accumulation contains the accumulated number of grids upstream to it for each grid using a recursive procedure (Mark, 1988). Fig. 5.7 shows the flow direction and flow accumulation algorithms performed on a synthetic DEM grid, and Fig. 5.8 shows the dialog of 'Flow Grids' module.

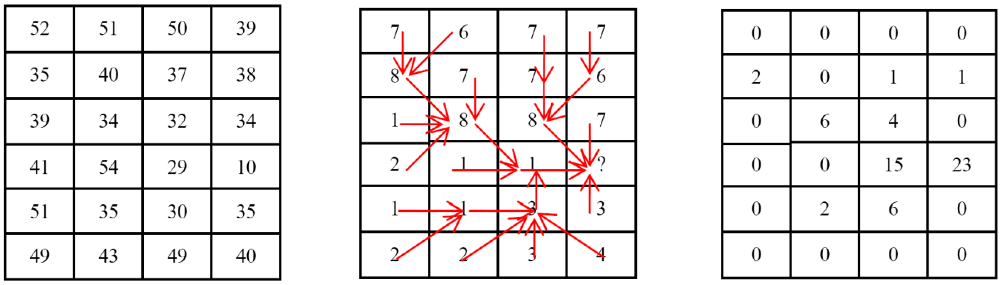


Figure 5.7 Flow direction and flow accumulation for a synthetic grid

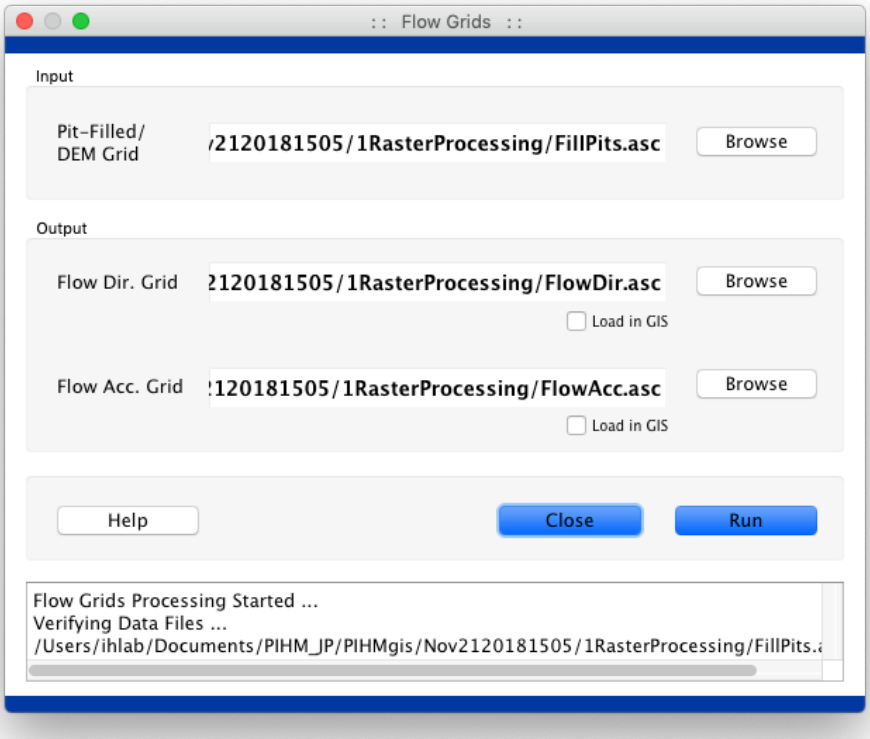


Figure 5.8 Flow Grids dialog

5.4 Stream Grids

'Stream Grids' module marks 1 for grids which has higher value of flow accumulation than the threshold (Fig. 5.9) . You can get threshold by press the 'Suggest Me' button, and you can write an integer value in the blank space yourself. The name of output files will contain the threshold value, so you can distinguish your trials easily.

0	0	0	0
1	0	0	0
0	1	1	0
0	0	1	1
0	1	1	0
0	0	0	0

Figure 5.9 Stream Grid for the synthetic grid

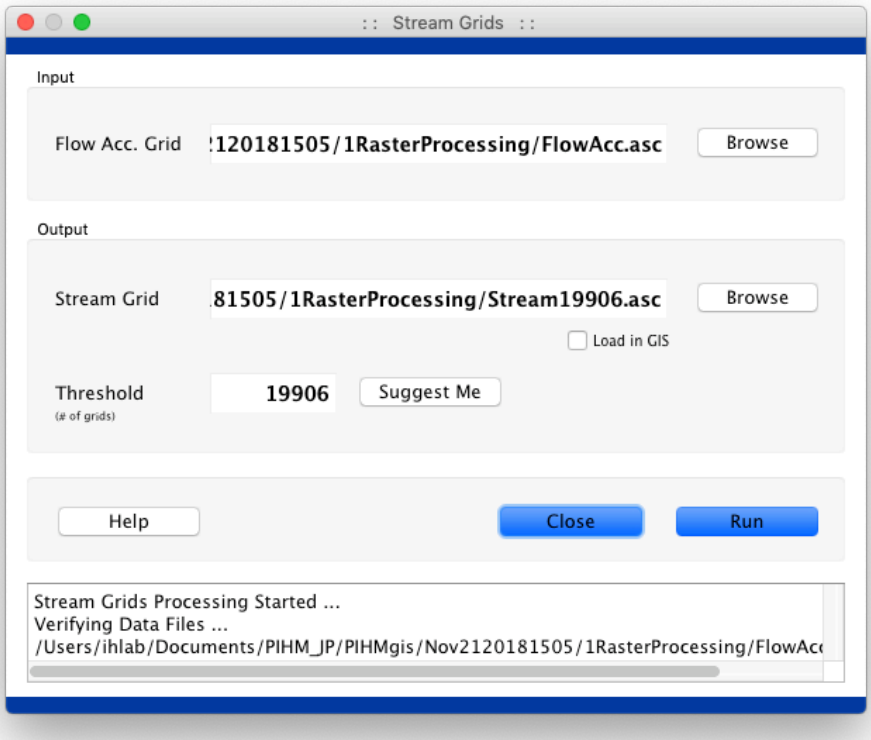


Figure 5.10 Stream Grids dialog

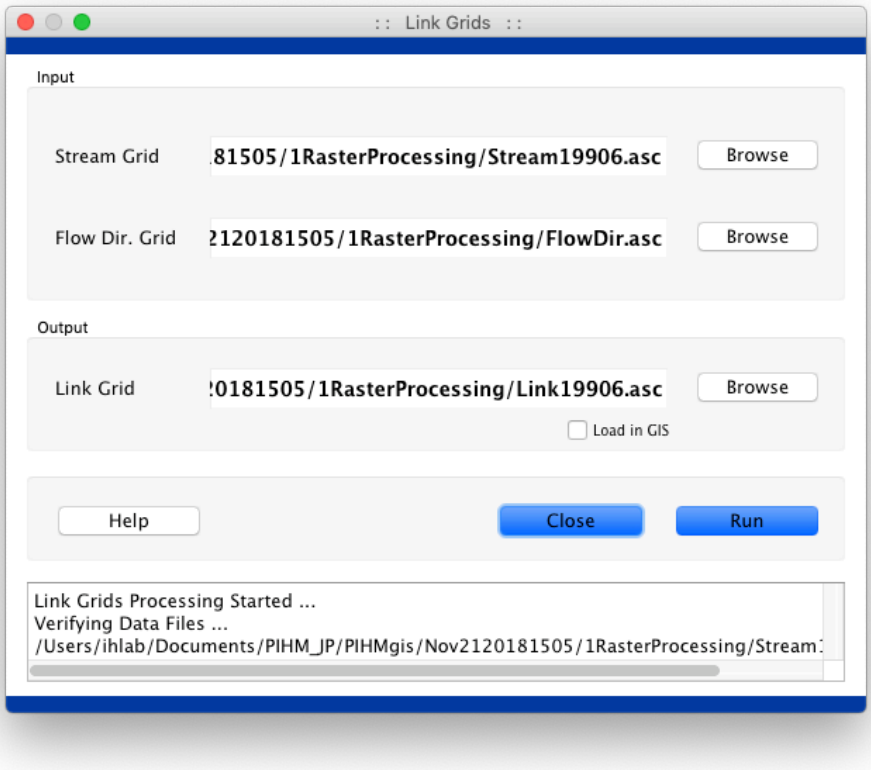
5.5 Link Grids

'Link Grids' module separates the stream grid segments at the junctions. Each link grid segment is assigned a unique integer value starting with 1, and rest of grid assumed No Data Value. Fig. 5.11 shows the link grid generated corresponding to the stream grid obtained in the previous section.

0	0	0	0
3	0	0	0
0	3	3	0
0	0	1	1
0	2	2	0
0	0	0	0

208
209 *Figure 5.11 Link Grid for the synthetic grid*

210



211
212 *Figure 5.12 Link Grids dialog*

5.6 Catchment Grids

All the grids draining to a particular stream segment are grouped into one catchment grid. Catchment grids are marked with integer numbers as same as link grid (Fig. 5.13).

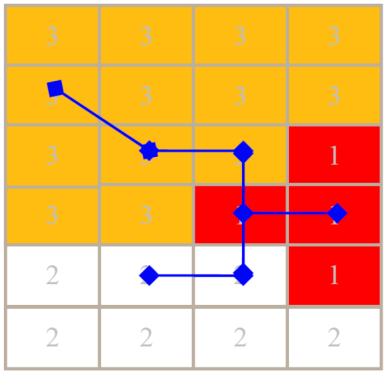


Figure 5.13 Catchment grid for the synthetic grid

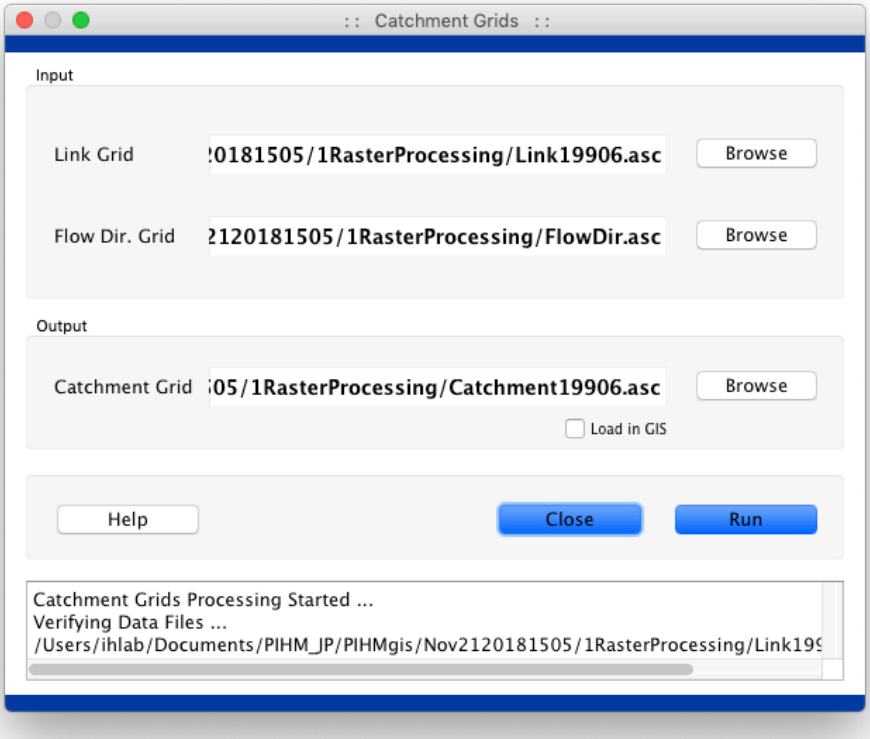


Figure 5.14 Catchment Grids dialog

5.7 Stream Polyline

Stream polyline is the drainage network for the region of interest obtained by the conversion of the link grid raster to a vector data (Fig.

5.15). Each link segment forms an individual stream segment and connected at the junction points. Flow direction is used to ensure that the segments are topologicallu correct.

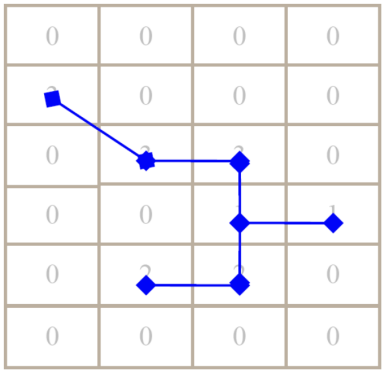


Figure 5.15 Stream Polyline for the synthetic grid

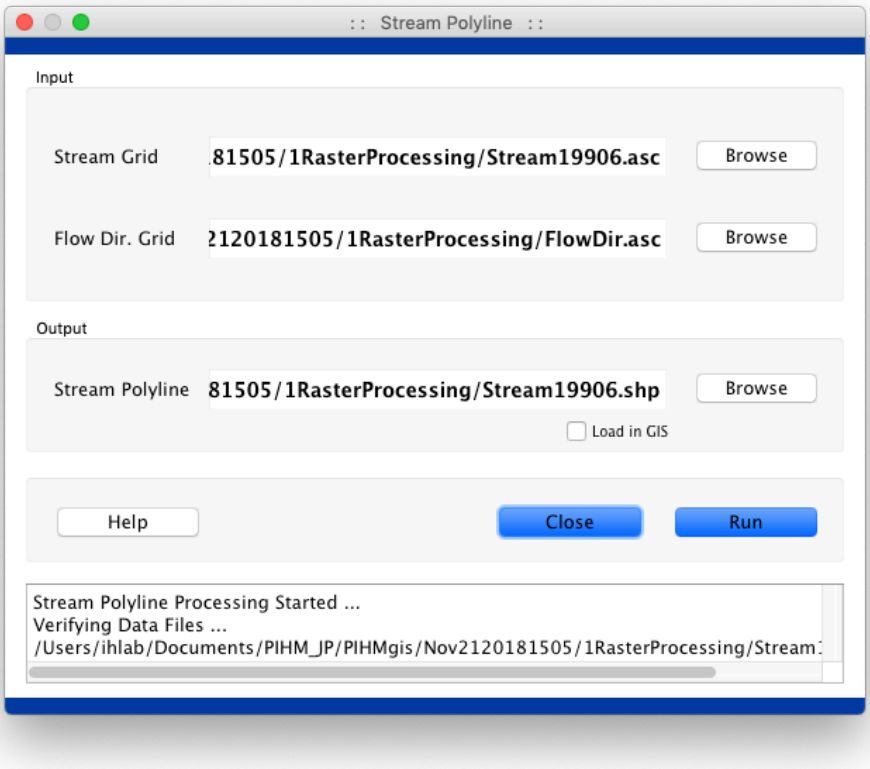


Figure 5.16 Stream Polyline dialog

5.8 Catchment Polygon

Catchment polygons is the vector representation of the catchment grid. Similar to the catchment grid, a catchment polygon bounds the region which has a single drainage outlet. Boundary of all catchment grids

draining to the same stream segment forms a unique catchment polygon.

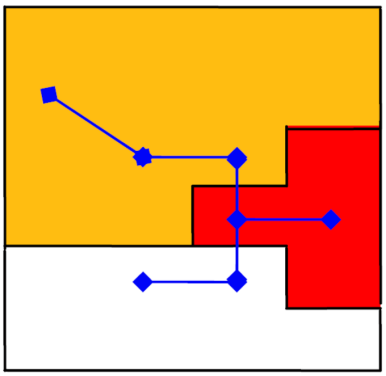


Figure 5.17 Catchment Polygon for the synthetic grid

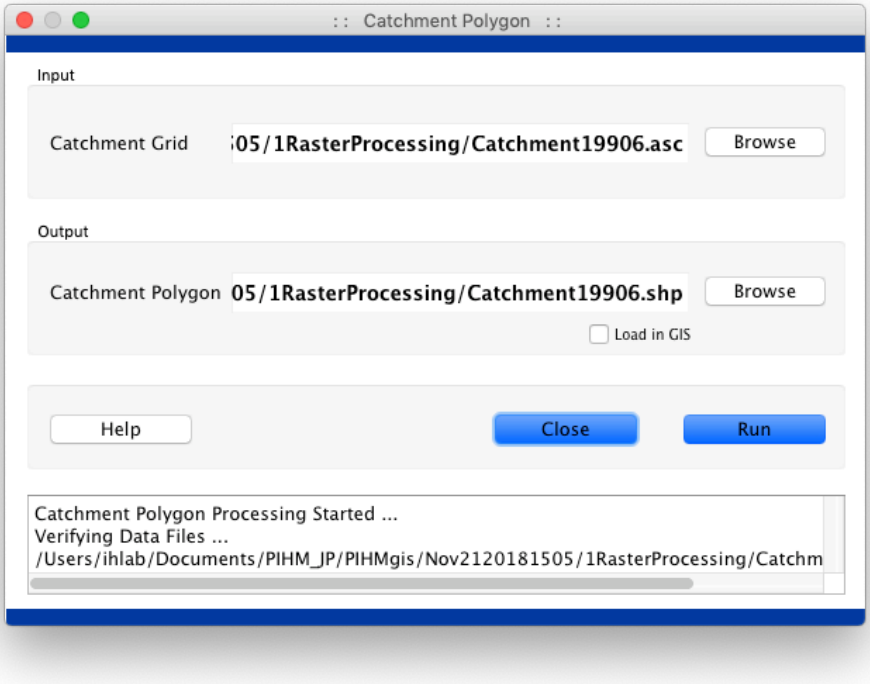


Figure 5.18 Catchment Polygon dialog

5.9 Summary

If you have performed all aforementioned steps, then you must have 6 raster files (fill pits, flow direction, flow accumulation, stream grid, link grid, and catchment grid) and 2 shape file groups (stream polyline and catchment polygon). These 2 shape file groups are the final results of 'Raster Processing', and they are automatically copied in the sub-folder of 'Vector Processing' by PIHMgis. Before moving on to the 'Vector Processing', checking the results is suggested. Obtained shape files are shown in Fig. 5.19.

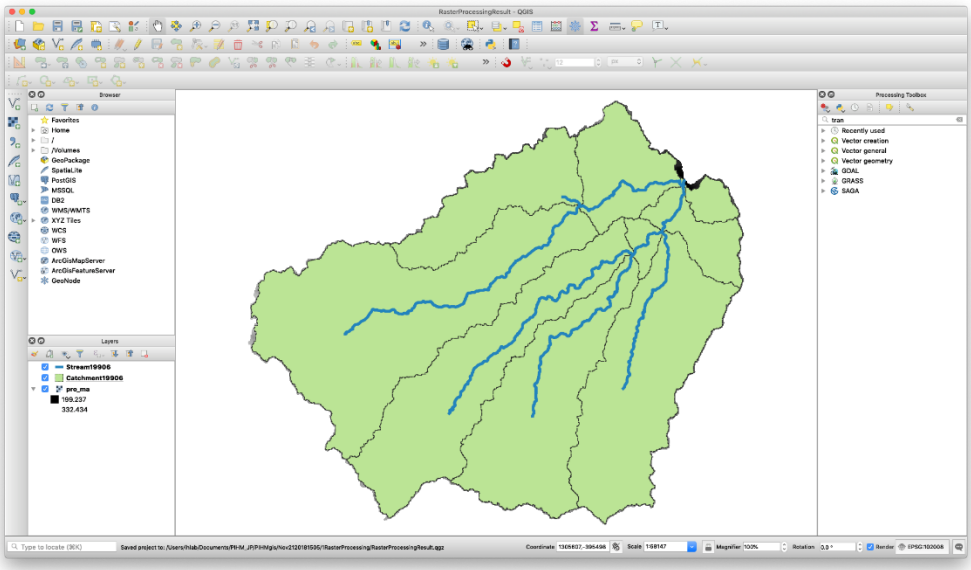


Figure 5.19 Stream polyline and catchment polygon represented in QGIS

6 Vector Processing

6.1 Dissolve Polygons

'Vector Processing' is a sequence of steps to "prepare" catchment polygon and stream polyline for 'Domain Decomposition'. Vector Processing includes dissolve, simplification, and splitting operation.

In 'Dissolve Polygons' module, delineated sub-catchments are dissolved to create a watershed bounday. The name of output file has "_dis" to distinguish easily.

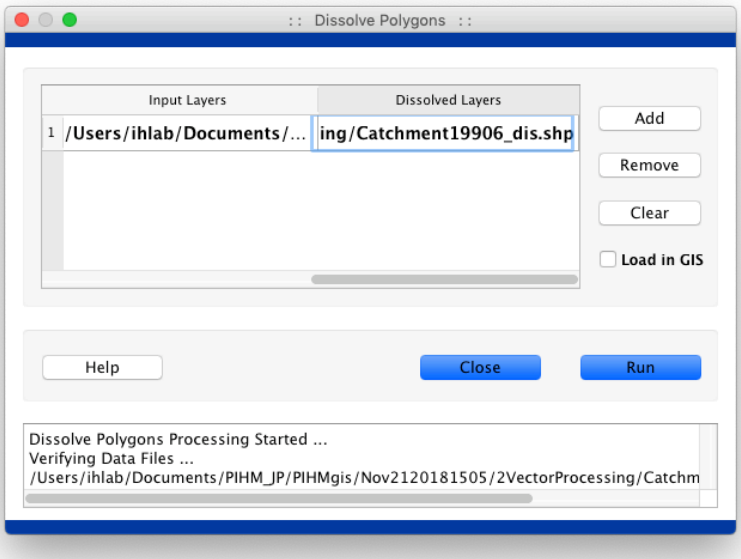


Figure 6.1 Dissolve Polygons dialog

6.2 Polygon to Polylines

'Polygon to Polylines' module converts the watershed polygon into a polyline file. Fig. 6.2 shows a schematic of the steps involved in the algorithm for 'Polygon to Polyline' module.

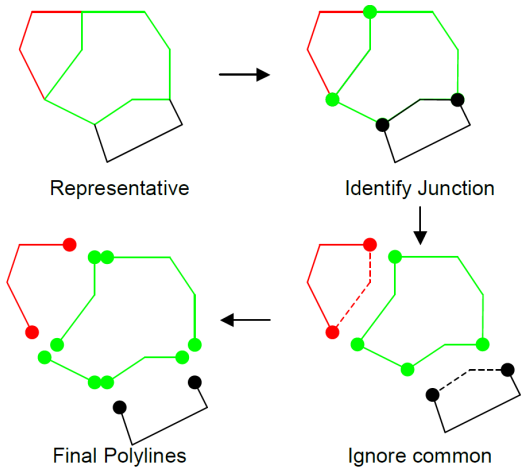


Figure 6.2 Algorithm for Polygon to Polyline

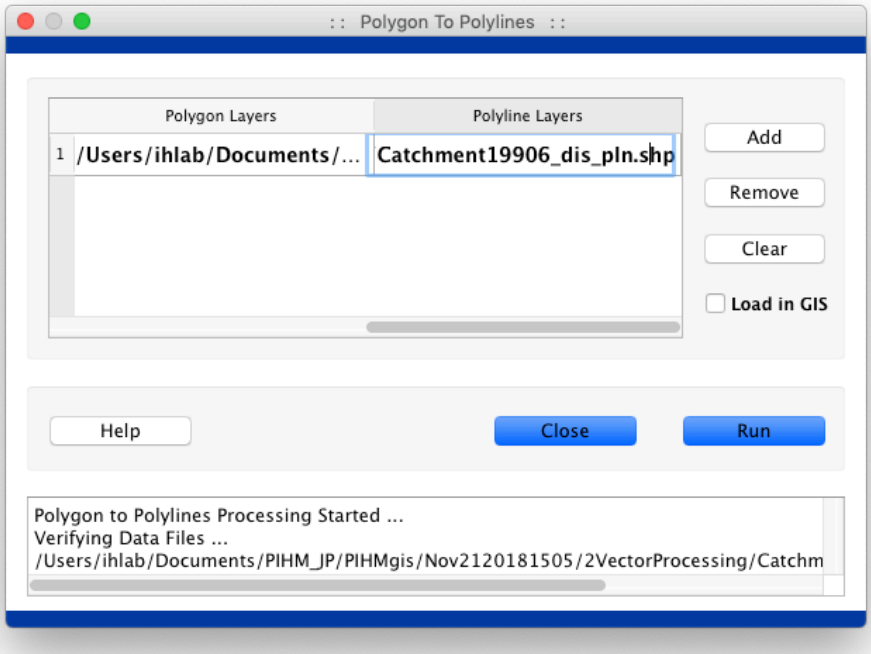


Figure 6.3 Polygon to Polylines dialog

6.3 Simplify Lines

To control the distribution of triangles and reduce the effects of quick fluctuations in the polyline file (obtained above) on triangle distribution, 'Simplify lines' operation is implemented. It removes fluctuations or extraneous bends using Douglas-Peucker algorithm (Fig. 6.4). Higher tolerance (m) makes polylines (catchment boundaries and streams) simpler.

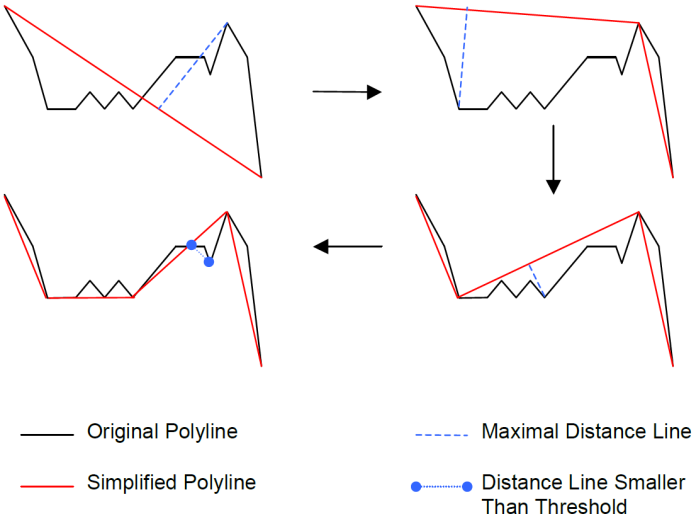


Figure 6.4 Algorithm for Simplification

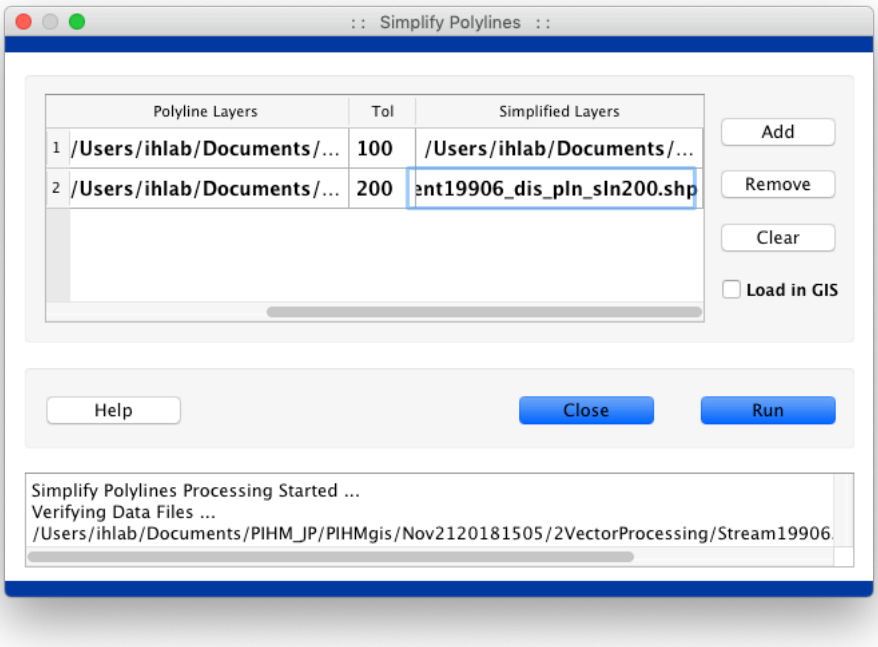


Figure 6.5 Simplify Polylines dialog

If you want to include lakes or wetlands as internal boundaries for domain discretization, you should include them in this step. That way they can be simplified as well. Note, different simplification tolerances can be provided to different files.

6.4 Polyline to Split Lines and modifications

For creating triangle elements in 'Domain Decomposition', all line segments should be considered as an individual feature object. 'Split Lines' module splits polylines at each vertex. 'Split Lines' module is described schematically in Fig. 6.6. Input has only one feature which is of type polyline. Output has three features which are of the type lines. The name of output file has "_xln" at the end to distinguish easily.

Figure 6.6 Schematically described split line operation

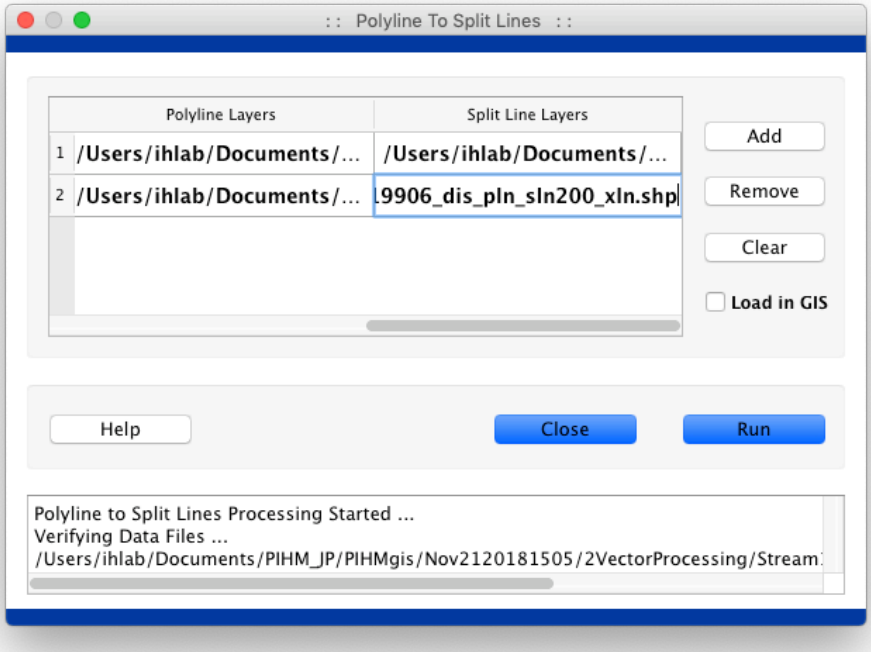


Figure 6.7 Polyline To Split Lines dialog

Before moving on to the next step, the result of 'Split Lines' should be checked. In Fig. 6.8, you may see the stream outlet (blue line) and catchment boundary (black line) don't meet each other (it is also possible that sometimes the stream may intersect the catchment boundary but not end at it). This problem is caused by simplification. To address this problem, 'editing' tool should be used to move a nearby catchment boundary vertex to the river outlet. The end point of stream outlet could be moved to the catchment boundary vertex, but in this case you have to make sure that the elevation at the moved location is lower than at the upstream stream vertex.

Fig. 6.8 shown the editing process in QGIS.

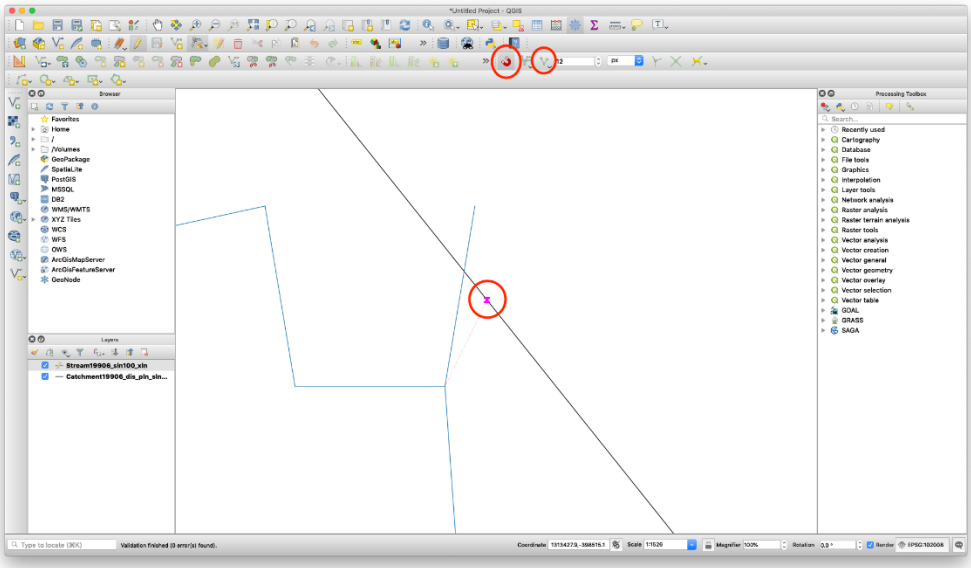


Figure 6.8 Modifying split stream and catchment boundary with snapping option

6.5 Vector Merge

'Vector Merge' is the final step of 'Vector Processing'. It merges stream and catchment boundary (and other internal boundaries if they exist), so that a merged shape file will be used in 'Domain Decomposition'. Fig. 6.10 shows the result of 'Vector Processing'.

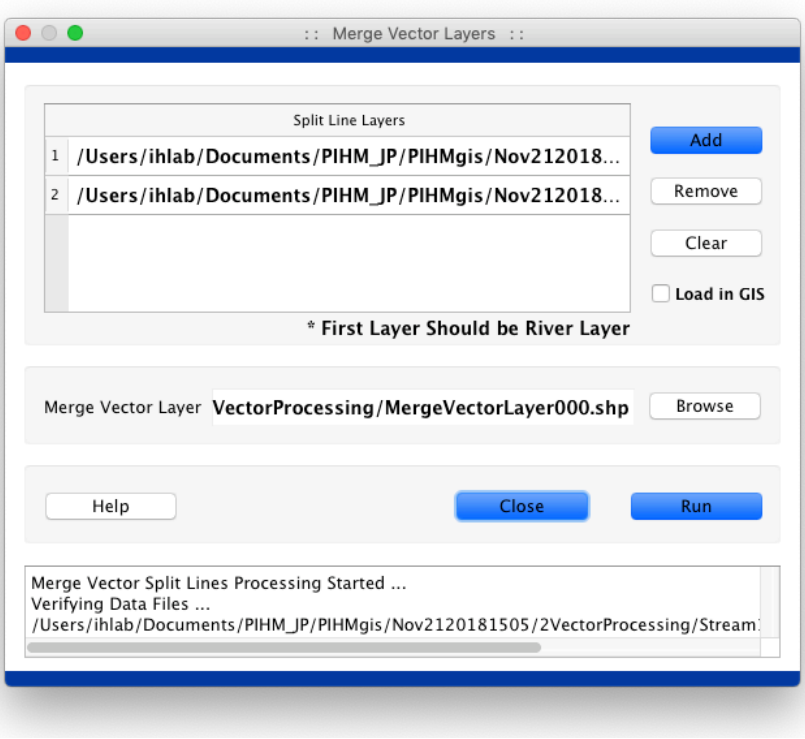


Figure 6.9 Merge Vector Layers dialog

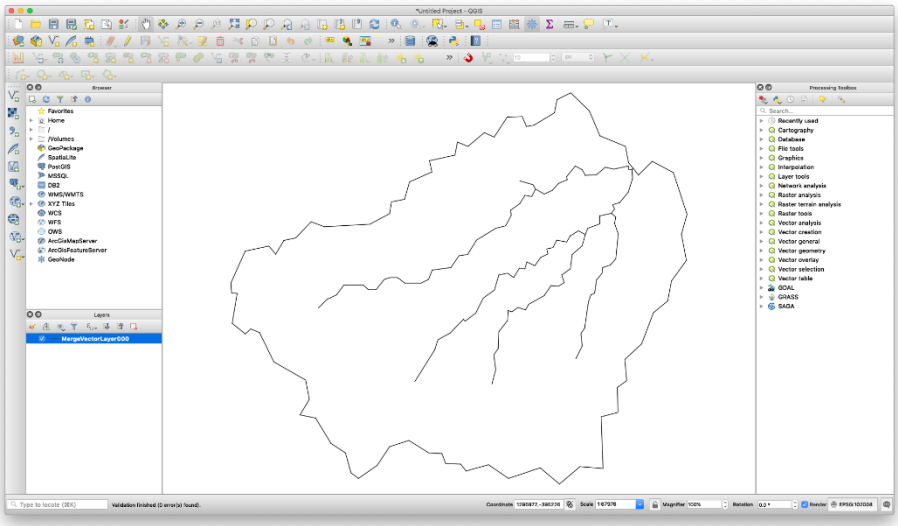


Figure 6.10 Split lines including stream line and catchment boundary

7 Domain Decomposition

'Domain Decomposition' is the process of creating mesh using internal and external boundary split lines (of stream, catchment boundary etc.). This is done using the following steps:

7.1 Read Topology

All the vector line information from the merged shape file is extracted into planer straight line graph (PSLG), and is saved in a '.poly' file.

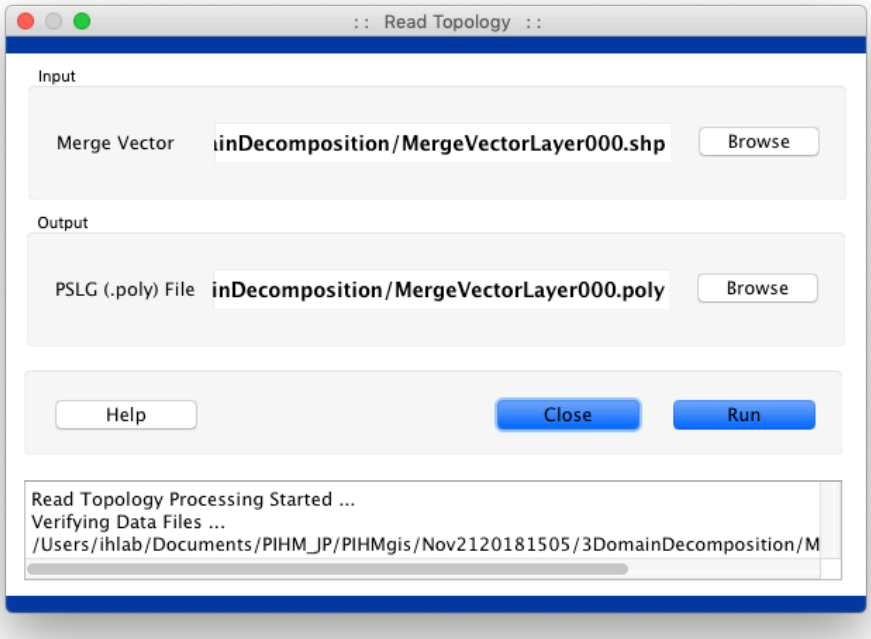


Figure 7.1 Read Topology dialog

7.2 Triangulation

The library 'TRIANGLE' developed by Shewchuk (2001) is one of the most efficient implementations of constrained Delaunay triangulation algorithm which provides numerous flexibility to the user regarding the number and distribution of triangles. It takes planner straight line graph (PSLG) as input. The algorithm works in such a way that it refines the Delaunay triangulation by inserting carefully placed vertices until the generated mesh meets a provided quality and size criterion.

'Angle (minimum angle)' and 'Area (maximum area)' constraint options are provided in PIHMgis interface. You may use 'Others' option to access other TRIANGLE program commands. These options are provided in Table 7.1. This step generates '.poly', '.node', '.ele', and '.neigh' files, which define the mesh.

In this tutorial, min. angle was set as 30 degree (to avoid sharp angle problem) and max. area was set as 1,000,000 m^2 to create small number of mesh grids to shorten the run time.

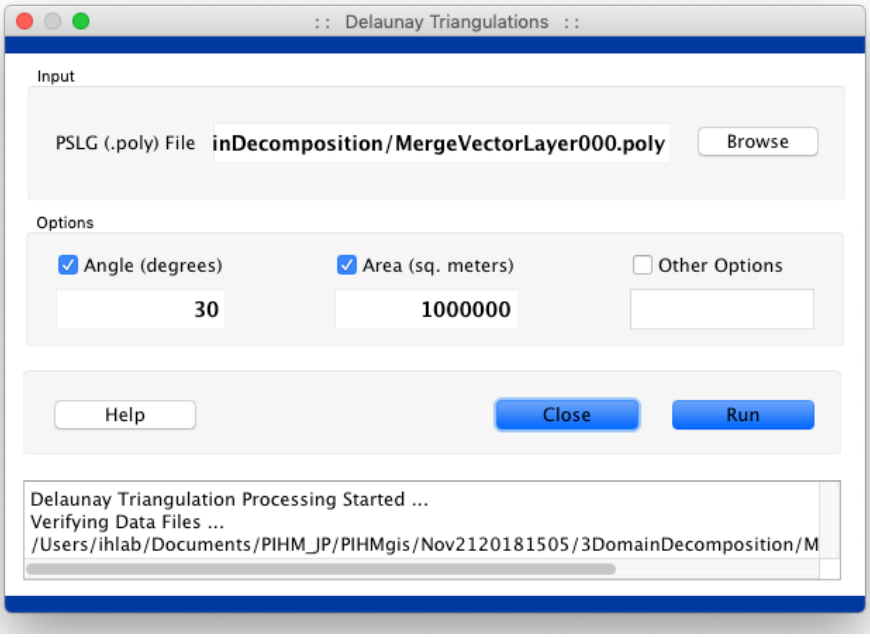


Figure 7.2 TRIANGULATION dialog

Table 7.1 TRIANGLE options (shaded options are provided respectively in dialog)

Options	Description
-p	Triangulates a Planar Straight Line Graph (.poly file).
-r	Refines a previously generated mesh.
-q	Quality mesh generation. A minimum angle may be specified.
-a	Applies a maximum triangle area constraint.
-u	Applies a user-defined triangle constraint
-A	Applies attributes to identify triangles in certain regions.
-c	Encloses the convex hull with segments.
-w	Weighted Delaunay triangulation.
-W	Regular triangulation (lower hull of a height field).
-j	Jettison unused vertices from output .node file.
-e	Generates an edge list.
-v	Generates a Voronoi diagram.
-n	Generates a list of triangle neighbors.
-g	Generates an .off file for Geomview.
-B	Suppresses output of boundary information.
-P	Suppresses output of .poly file.
-N	Suppresses output of .node file.
-E	Suppresses output of .ele file.
-I	Suppresses mesh iteration numbers.
-O	Ignores holes in .poly file.
-X	Suppresses use of exact arithmetic.
-z	Numbers all items starting from zero (rather than one).

-o2	Generates second-order sub-parametric elements.
-Y	Suppresses boundary segment splitting.
-S	Specifies boundary segment splitting.
-L	Uses equatorial circles, not equatorial lenses.
-i	Uses incremental method, rather than divided-and-conquer.
-F	Uses Fortune's sweepline algorithm, rather than d-and-c.
-l	Uses vertical cuts only, rather than alternating cuts.
-s	Force segments into mesh by splitting (instead of using CDT).
-L	Uses Ruppert's diametral spheres, not diametral lenses.
-C	Check consistency of final mesh.
-Q	Quiet: No terminal output except errors.
-V	Verbose: Detailed information on what I'm doing.
-h	Help: Detailed instructions for Triangle.

7.3 Tin Shape Layer

In this step, `.ele' file and `.node' files are used to generate TIN shape file as shown in Fig. 7.4. Target watershed was discretized into 969 triangles.

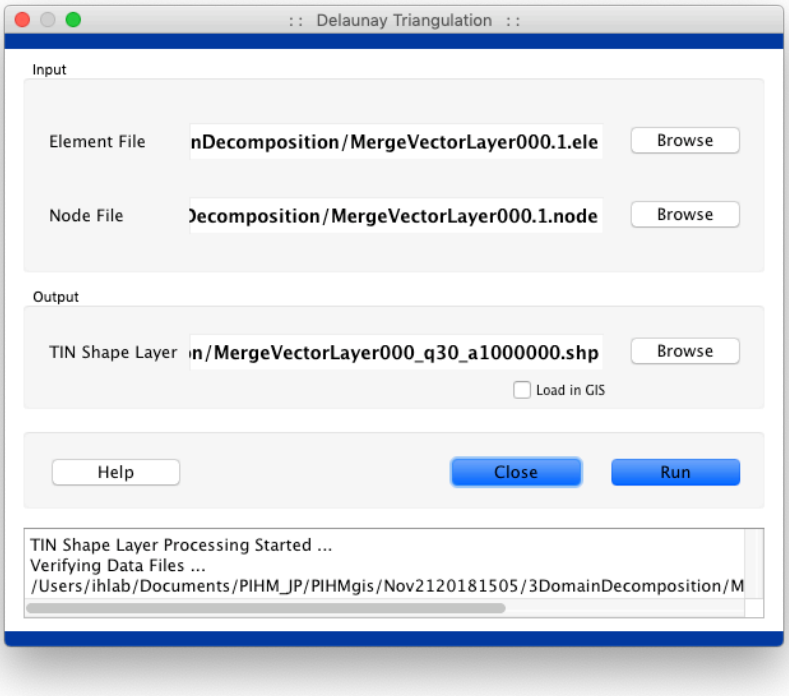


Figure 7.3 Tin Shape Layer dialog

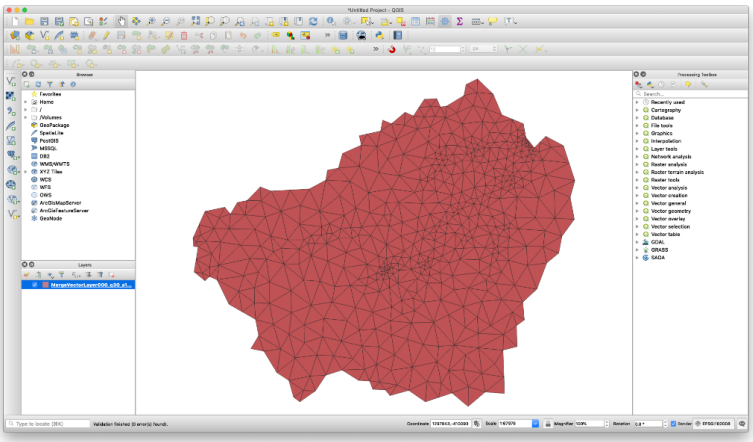


Figure 7.4 Created triangle mesh grids

8 Pre-processing raw data

8.1 Introduction

Before we can populate the data on the mesh using the 'Data Model Loader', pre-processing of raw data has to be done. Essentially the goal is to create class grid files for physiographic and meterological properties. These files will eventually be used to generate PIHM attribute (.att) input file. Correspondingly, the parameters associated with different classes have to be defined. These parameter information are used to generate PIHM input files such as '.soil', '.geol', '.lc', and '.forc'. Raw data used for pre-processing include distributed soil (and geology), land cover, and forcing data were used.

It is imperative that the raw data used here fully covers the watershed. Users are advised to use a buffer shape file that is larger by around 2 times the size of the coarsest grid . As shown in Fig. 8.1, buffer shape file is large enough to cover mesh shape file. This buffer shape file will be used to clip raster input layers.

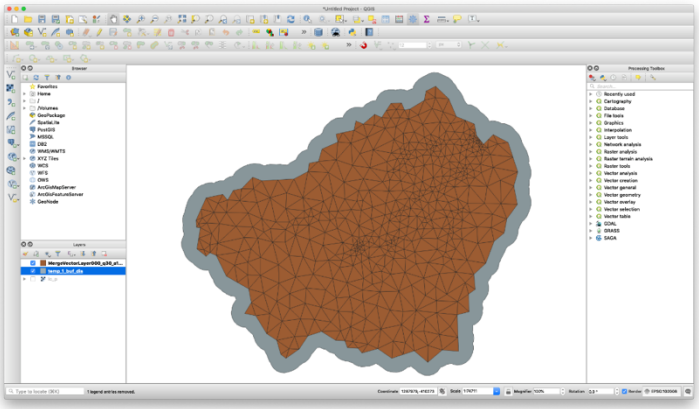


Figure 8.1 Mesh shape file and buffer shape file

8.2 Soil Data

Before we move forward, it is to be noted that an alternative to performing the following processing using the SSURGO data is to manually generate .soil file for the soil cover map of your choice.

Easiest way to download and extract soil attributed data (especially SSURGO) is using 'Soil Data Viewer[8]' that is a plug-in application of ArcGIS, and 'Soil Data Development Toolbox[9]' that is a library of ArcGIS. The 'Soil Data Viewer' links shape file and database of SSURGO to extract relevant soil parameters. The 'Soil Data Development Toolbox' includes downloading, mapping, data management, and reporting tools. Even though this tutorial focused on using QGIS in other sections as it is open-source, ArcGIS is used in this section. If ArcGIS is not available, you should extract parameters from database on your own.

This tutorial downloaded soil data from 'Web Soil Survey (WSS)' website (Fig. 8.2). You can use 'AOI (Area of Interest)' to download the shape file of the target watershed.

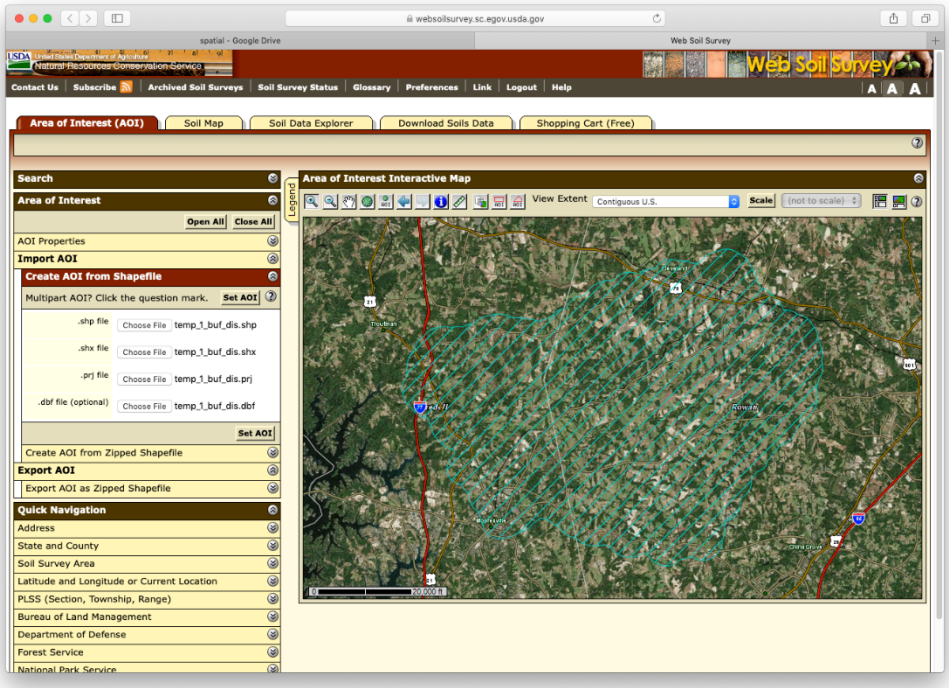


Figure 8.2 'Web Soil Survey' website and AOI of buffer shape file

You can download soil data as shown in Fig. 8.3. After selecting AOI, press the 'Download Soils Data' tab and press 'Create Download Link'

[8] https://www.nrcs.usda.gov/wps/portal/nrcs/detailfull/soils/survey/geo/?cid=nrcs142p2_053620
[9] https://www.nrcs.usda.gov/wps/portal/nrcs/detail/soils/survey/geo/?cid=nrcs142p2_053628

located below. Then, you will get a hyperlink 'wss_aoi_(date)_(time).zip'. This compressed file includes the shape files and MS Access database file. Downloaded soil data shape files ('soilmu_a_aoi.shp' which contains 'MUKEY') is shown as Fig. 8.4.

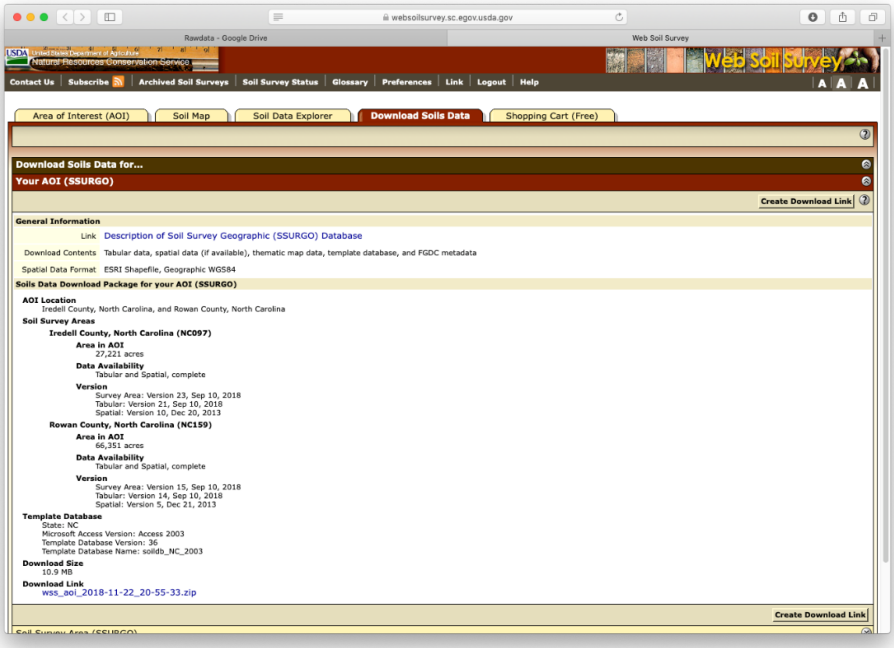


Figure 8.3 Download link of soil data shape file on the WSS website

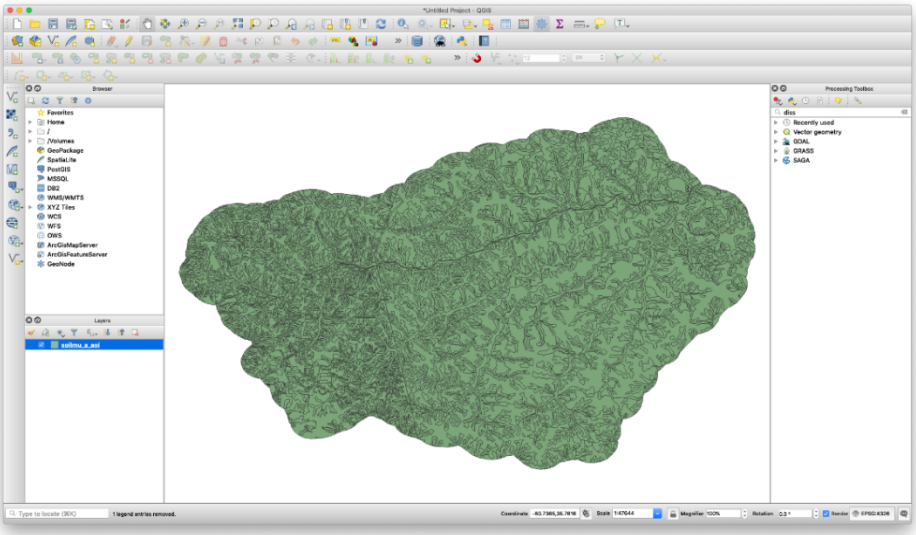


Figure 8.4 SSURGO shape file for target watershed

For generating soil (.soil) and geology (.geol) input data in sections 9.5 and 9.6, four parameters for each MUKEY should be extracted from the SSURGO database. In this regard, first open the MS Access database

file (‘.mdb’) that is included in the compressed file downloaded from the WSS website. When the Access database file is opened the first time, a dialog shown in Fig. 8.5 appears. Fill the path of ‘tabular’ folder in the blank space. It allows Soil Data Viewer to use SSURGO database.

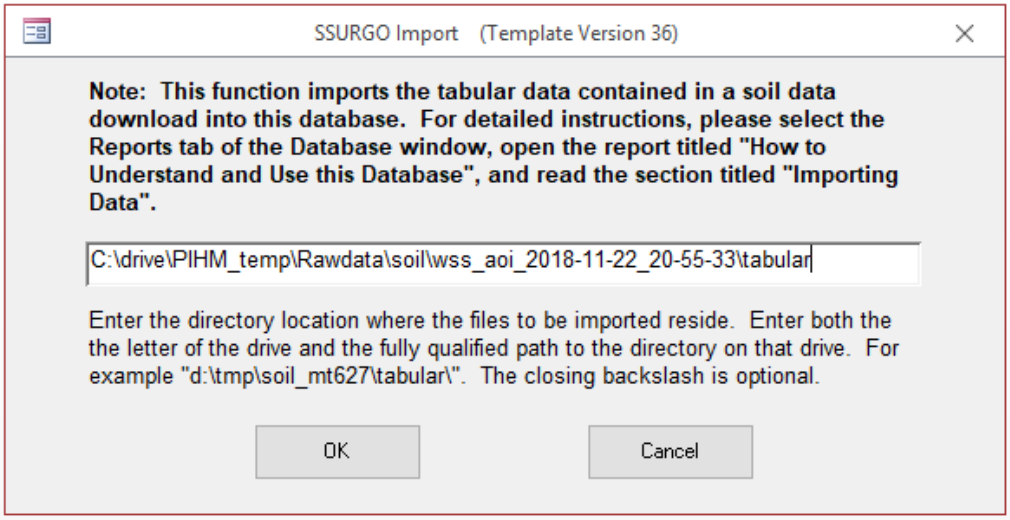


Figure 8.5 SSURGO import dialog for opening '.mdb' database

While opening AOI shape file, when you turn on the Soil Data Viewer, you will see a dialog shown in Fig. 8.6. Note that, you should use the downloaded shape file here with no modifications. Modified or pre-processed shape file will generate an error.

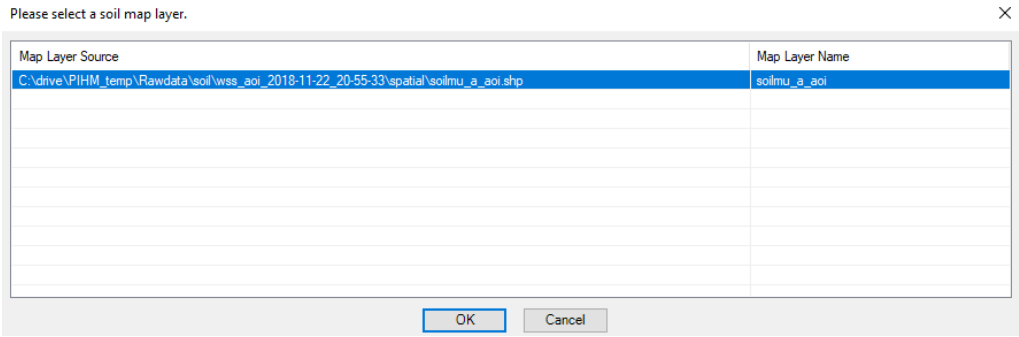


Figure 8.6 Dialog for selecting soil map layer for using Soil Data Viewer

When the Soil Data Viewer is launched , you will see a dialog shown in Fig. 8.7. There are three lines below. At the right side of ‘Synchronization Status’, there is a green box. If your ‘Map Layer’ and ‘Database (.mdb)’ are not matched (the extent of database can be larger than map layer), the box goes to yellow or red. Then, you cannot use Soil Data Viewer, or result will have errors. After checking green box, you can create layers that contain the soil parameter that you select. Fig. 8.8 shows the result of extracting soil parameter from SSURGO

using Soil Data Viewer. You should specify several options in the dialog of Fig. 8.7 while extracting the layer in Fig. 8.8. Most important is 'Layer Option' in the 'Rating Options' tab. You can specify exact depth range, or choose surface layer, but this tutorial chose all layers.

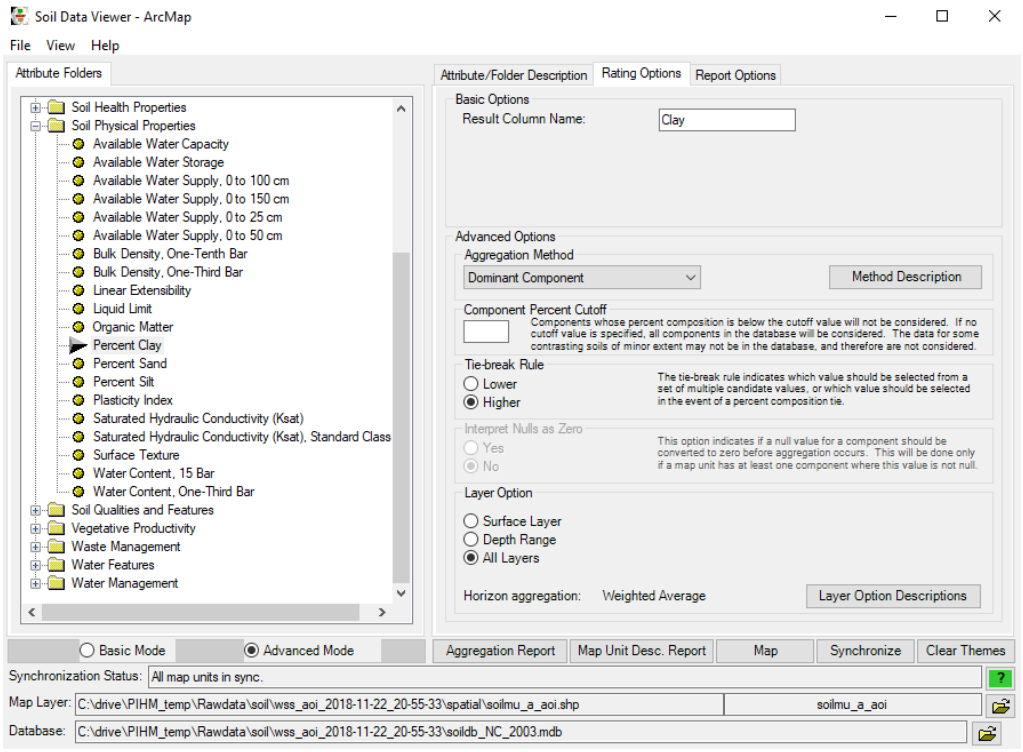


Figure 8.7 Soil Data Viewer dialog

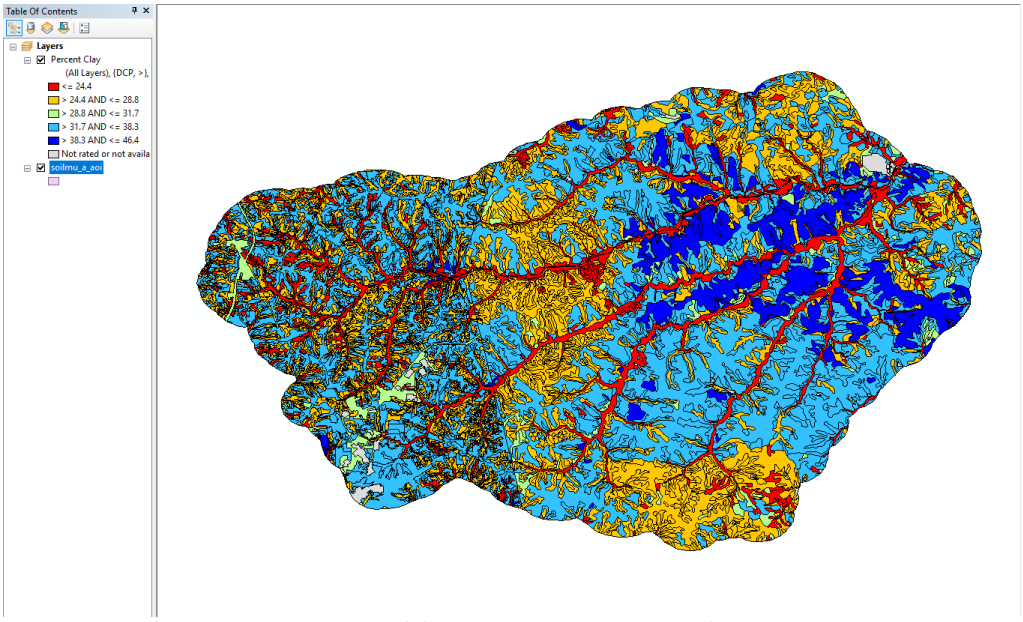


Figure 8.8 Clay percent extracted from SSURGO using Soil Data Viewer

The result (as shown in Fig. 8.8) is a kind of temporary polygon shape file. You need to merge it with target region shape file to store parameter values as shown in Fig. 8.9. You need to dissolve this shape polygon as shown in Fig. 8.10. At least, MUKEY and parameter values should be left as is. After sorting based on MUKEY (Fig. 8.11) and inserting ID column (Fig. 8.12), you can make the shape file for specific soil parameter.

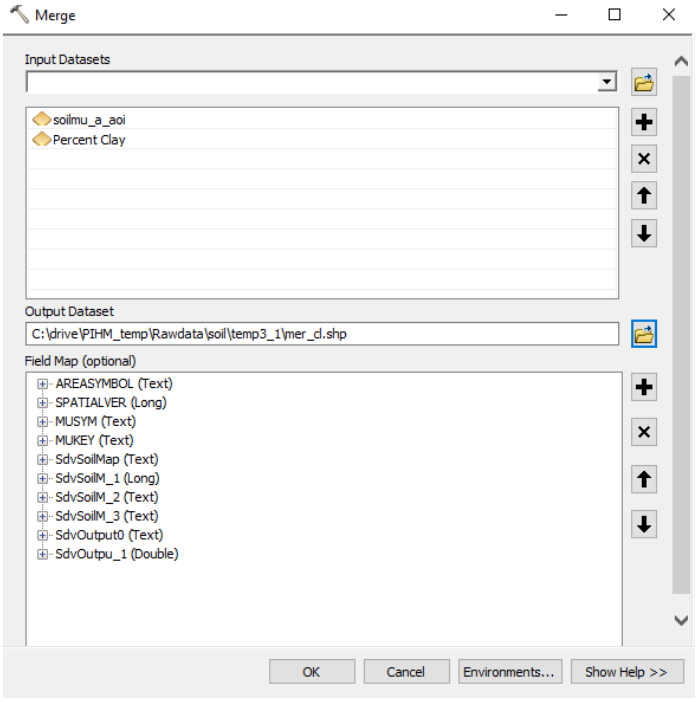
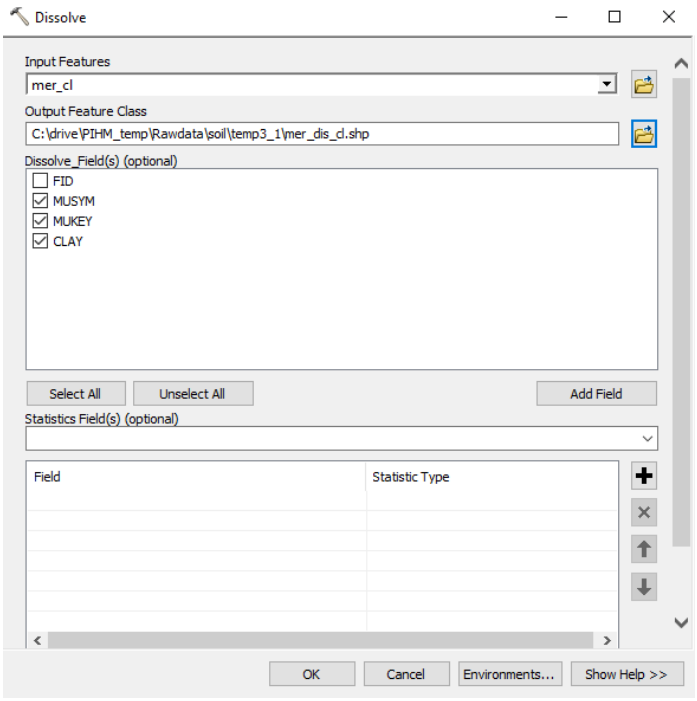
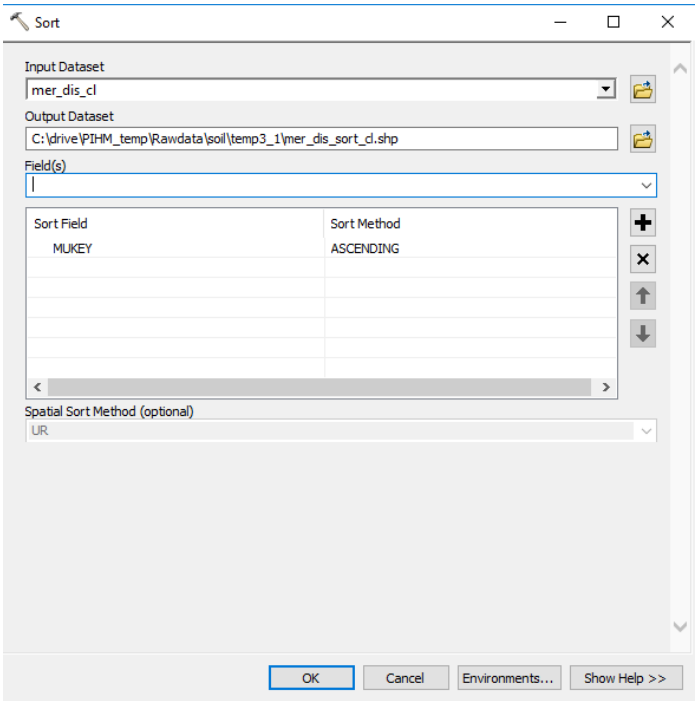


Figure 8.9 Merging dialog of ArcMap for clay percent of each feature



476
477 *Figure 8.10 Dissolving dialog of ArcMap for clay percent of each feature*

478



479
480 *Figure 8.11 Sorting dialog of ArcMap for clay percent of each MUKEY*

481

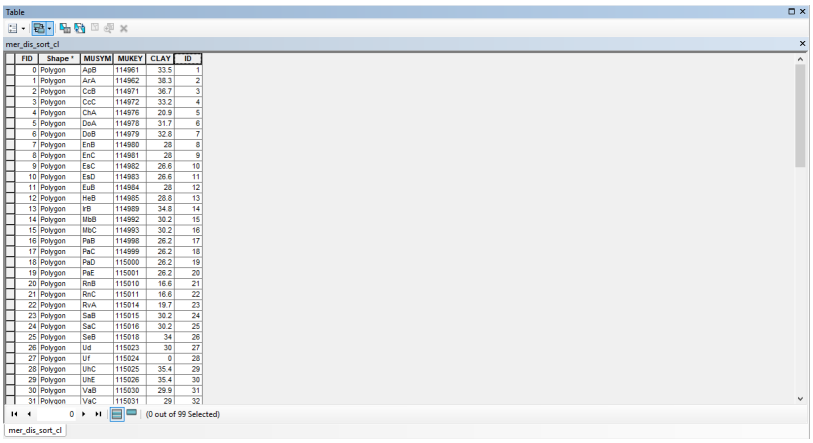


Figure 8.12 Attribute table of clay percent shape file after merging, dissolving, sorting, and inserting ID column

For creating '.att' file and allocating soil types to each triangular element, you need the '.asc' raster file with id number that indicates each soil type. First, you can make raster file based on the ID column as shown in Fig. 8.13. As PIHM needs spatial data to be in the projected coordinate system, appropriate projection transformations should be performed (Fig. 8.14). Finally, as shown in Fig. 8.15, '.asc' file should be generated, because PIHMgis requires '.asc' format to read the raster. This '.asc' raster file based on id will be used in section 9.3.

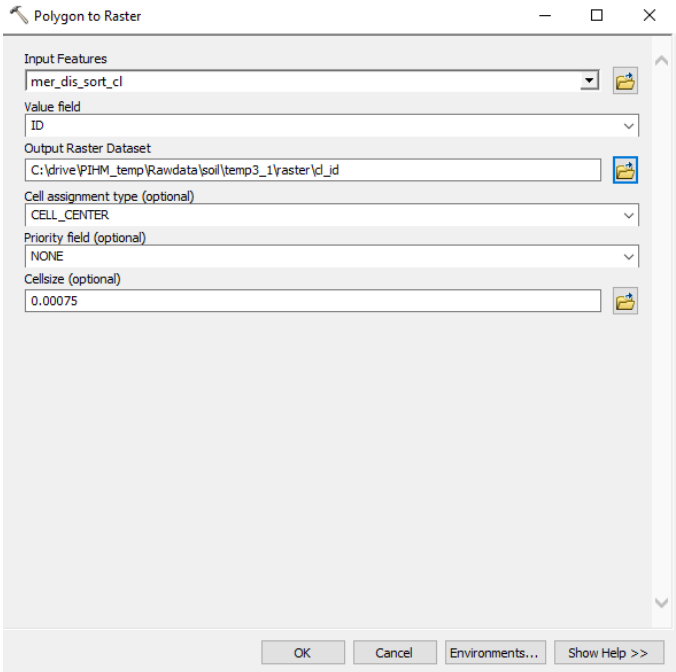


Figure 8.13 Polygon to Raster dialog for generating raster file based on ID column

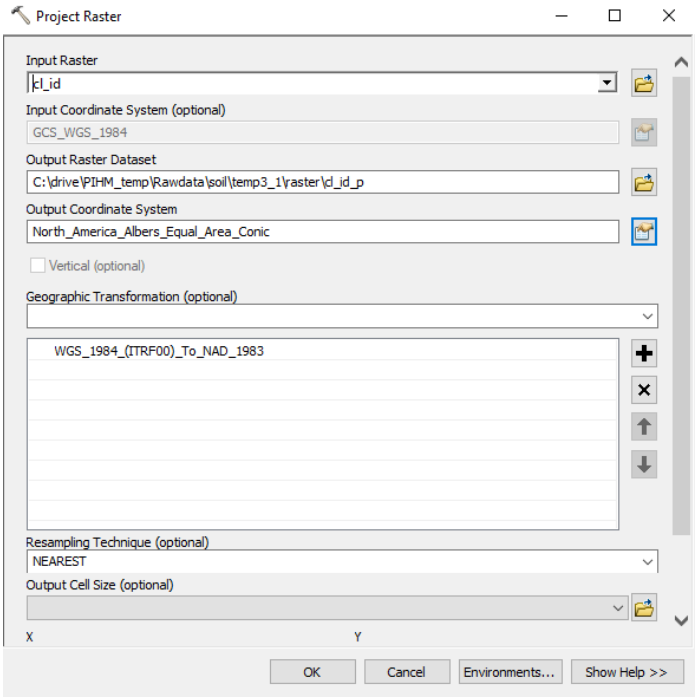


Figure 8.14 Project Raster dialog for changing projection of soil ID raster

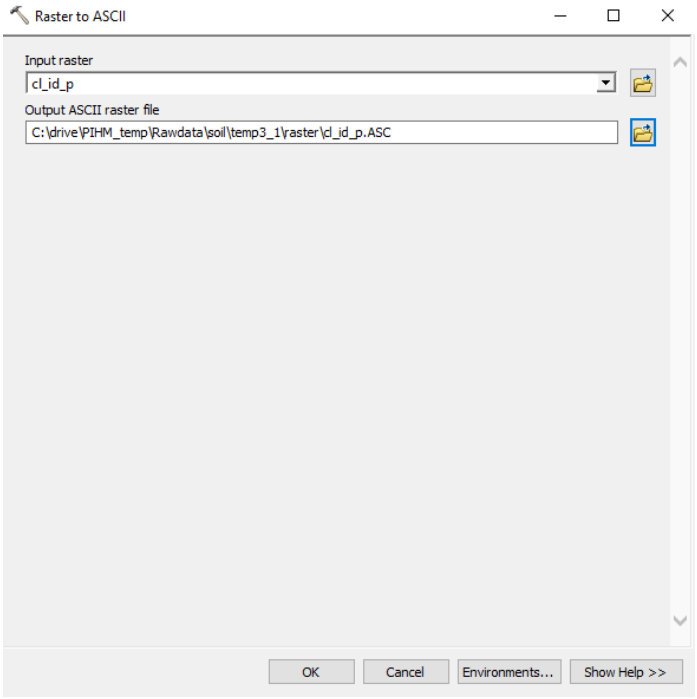


Figure 8.15 Raster to ASCII dialog to change format into '.asc' to use in PIHMgis

You may need several soil parameters. For example, PIHM needs clay percent, silt percent, organic matter, and bulk density. Shape files similar to one shown in Fig. 8.12 for each soil parameter can be generated. It is to be noted that there may be some grids where data

does not exist. These could be locations covered by water perennially. For these grids, an option is use majority filter to fill in the parameter values.

8.3 Land Cover Data

Before we move forward, it is to be noted that an alternative to performing pre-processing on land cover data is to manually generate .lc file for the land use/land cover classification map of your choice.

Land cover data can also be downloaded at the TNM Download v1.0 (Fig. 8.16). This tutorial used NLCD 2011 for North Carolina (Fig. 8.17).

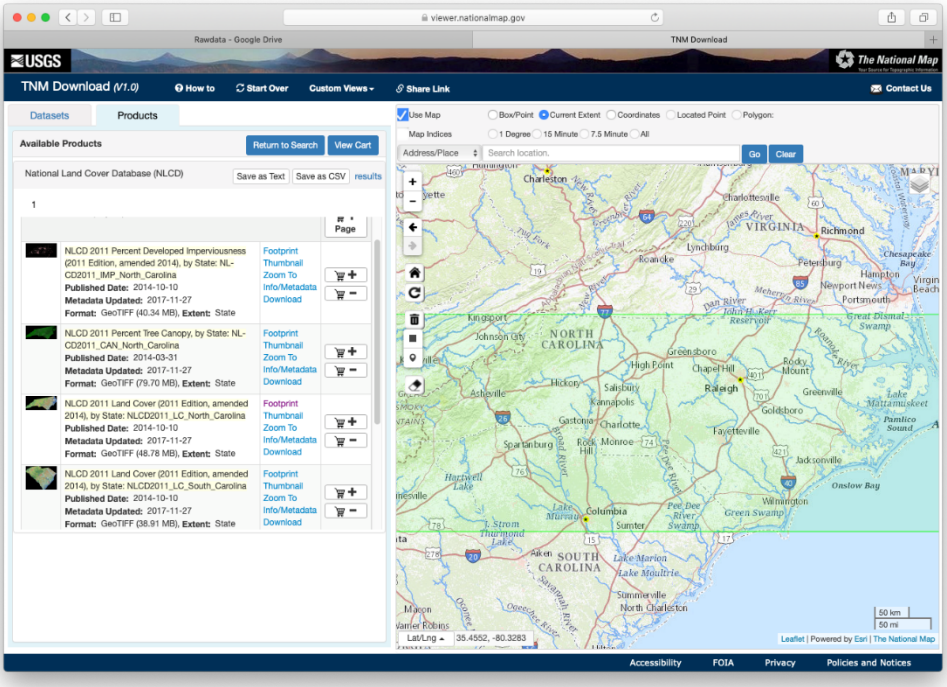


Figure 8.16 TNM Download website for Land Cover (http://viewer.nationalmap.gov)

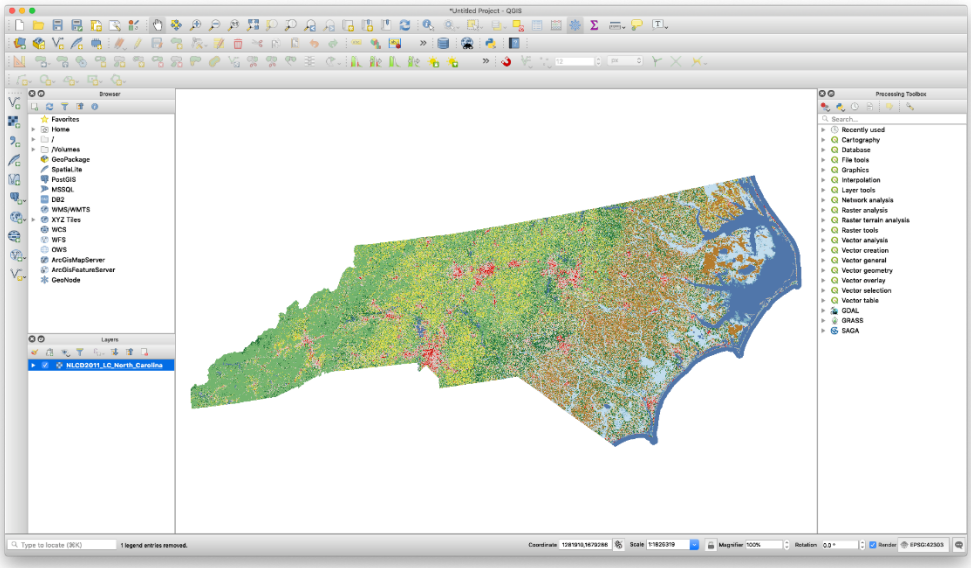


Figure 8.17 NLCD 2011 for North Carolina

Before moving forward, have a quick look at Figs. 8.18 and 8.19. Fig. 8.18 shows the legend of NLCD 2011. Fig. 8.19 shows the legend of UMD Land Cover Classification. Since PIHMgis generates PIHM land cover parameter file (.lc) file based on NLCD, you will have to accordingly reclassify if you are using other land cover data such as the UMD Land Cover Classification (UMDLCC).

Class\ Value	Classification Description
Water	
11	**Open Water**- areas of open water, generally with less than 25% cover of vegetation or soil.
12	**Perennial Ice/Snow**- areas characterized by a perennial cover of ice and/or snow, generally greater than 25% of total cover.
Developed	
21	**Developed, Open Space**- areas with a mixture of some constructed materials, but mostly vegetation in the form of lawn grasses. Impervious surfaces account for less than 20% of total cover. These areas most commonly include large-lot single-family housing units, parks, golf courses, and vegetation planted in developed settings for recreation, erosion control, or aesthetic purposes.
22	**Developed, Low Intensity**- areas with a mixture of constructed materials and vegetation. Impervious surfaces account for 20% to 49% percent of total cover. These areas most commonly include single-family housing units.
23	**Developed, Medium Intensity**- areas with a mixture of constructed materials and vegetation. Impervious surfaces account for 50% to 79% of the total cover. These areas most commonly include single-family housing units.
24	**Developed High Intensity**-highly developed areas where people reside or work in high numbers. Examples include apartment complexes, row houses and commercial/industrial. Impervious surfaces account for 80% to 100% of the total cover.
Barren	
31	**Barren Land (Rock/Sand/Clay)** - areas of bedrock, desert pavement, scarps, talus, slides, volcanic material, glacial debris, sand dunes, strip mines, gravel pits and other accumulations of earthen material. Generally, vegetation accounts for less than 15% of total cover.
Forest	
41	**Deciduous Forest**- areas dominated by trees generally greater than 5 meters tall, and greater than 20% of total vegetation cover. More than 75% of the tree species shed foliage simultaneously in response to seasonal change.
42	**Evergreen Forest**- areas dominated by trees generally greater than 5 meters tall, and greater than 20% of total vegetation cover. More than 75% of the tree species maintain their leaves all year. Canopy is never without green foliage.
43	**Mixed Forest**- areas dominated by trees generally greater than 5 meters tall, and greater than 20% of total vegetation cover. Neither deciduous nor evergreen species are greater than 75% of total tree cover.
Shrubland	
51	**Dwarf Scrub**- Alaska only areas dominated by shrubs less than 20 centimeters tall with shrub canopy typically greater than 20% of total vegetation. This type is often co-associated with grasses, sedges, herbs, and non-vascular vegetation.
52	**Shrub/Scrub**- areas dominated by shrubs; less than 5 meters tall with shrub canopy typically greater than 20% of total vegetation. This class includes true shrubs, young trees in an early successional stage or trees stunted from environmental conditions.
Herbaceous	
71	**Grassland/Herbaceous**- areas dominated by gramanoid or herbaceous vegetation, generally greater than 80% of total vegetation. These areas are not subject to intensive management such as tilling, but can be utilized for grazing.
72	**Sedge/Herbaceous**- Alaska only areas dominated by sedges and forbs, generally greater than 80% of total vegetation. This type can occur with significant other grasses or other grass like plants, and includes sedge tundra, and sedge tussock tundra.
73	**Lichens**- Alaska only areas dominated by fruticose or foliose lichens generally greater than 80% of total vegetation.
74	**Moss**- Alaska only areas dominated by mosses, generally greater than 80% of total vegetation.
Planted/Cultivated	
81	**Pasture/Hay**-areas of grasses, legumes, or grass-legume mixtures planted for livestock grazing or the production of seed or hay crops, typically on a perennial cycle. Pasture/hay vegetation accounts for greater than 20% of total vegetation.
82	**Cultivated Crops** -areas used for the production of annual crops, such as corn, soybeans, vegetables, tobacco, and cotton, and also perennial woody crops such as orchards and vineyards. Crop vegetation accounts for greater than 20% of total vegetation. This class also includes all land being actively tilled.
Wetlands	
90	**Woody Wetlands**- areas where forest or shrubland vegetation accounts for greater than 20% of vegetative cover and the soil or substrate is periodically saturated with or covered with water.
95	**Emergent Herbaceous Wetlands**- Areas where perennial herbaceous vegetation accounts for greater than 80% of vegetative cover and the soil or substrate is periodically saturated with or covered with water.

Figure 8.18 Legend of NLCD 2011 (https://www.mrlc.gov/data/legends/national-land-cover-database-2011-nlcd2011-legend)

UMD Land Cover Classification

Data Access

- Download via Search and Preview Tool (ESDI)
- Download via web page with links to FTP Server

Overview

The University of Maryland Department of Geography generated this global land cover classification collection in 1998. Imagery from the AVHRR satellites acquired between 1981 and 1994 were analyzed to distinguish fourteen land cover classes. This product is available at three spatial scales: 1 degree, 8 kilometer and 1 kilometer pixel resolutions.

Code Values for 1km and 8km data

Value	Label	RGB Red	RGB Green	RGB Blue
0	Water	068	079	137
1	Evergreen Needleleaf Forest	001	100	000
2	Evergreen Broadleaf Forest	001	130	000
3	Deciduous Needleleaf Forest	151	191	071
4	Deciduous Broadleaf Forest	002	220	000
5	Mixed Forest	000	255	000
6	Woodland	146	174	047
7	Wooded Grassland	220	206	000
8	Closed Shrubland	255	173	000
9	Open Shrubland	255	251	195
10	Grassland	140	072	009
11	Cropland	247	165	255
12	Bare Ground	255	199	174
13	Urban and Built	000	255	255

Figure 8.19 Legend of UMD Land Cover Classification (http://glcf.umd.edu/data/landcover/)

8.4 Forcing Data

The pre-processing procedure of forcing data contains three steps: downloading the data, generating spatial reference file, and reformatting downloaded data into '.forc' format. This section introduces the first two steps, and final step will be discussed in section 9.12.

In this tutorial, NLDAS 2 forcing data is used. The official full name is 'NLDAS Primary Forcing Data L4 Hourly 0.125 x 0.125 degree V002'. Users can use other climate or weather data for input. More information about the NLDAS-2 product is available in README document at https://hydro1.gesdisc.eosdis.nasa.gov/data/NLDAS/README.NLDAS2.pdf. Fig. 8.20 shows NASA GES DISC website (https://disc.gsfc.nasa.gov/datasets/NLDAS_FORA0125_H_V002/summary) that users can download NLDAS data from. In 'Data Access', there are several ways to download serial climate data. This tutorial used linux server, but users can use any other method explained in website (https://disc.gsfc.nasa.gov/data-access).

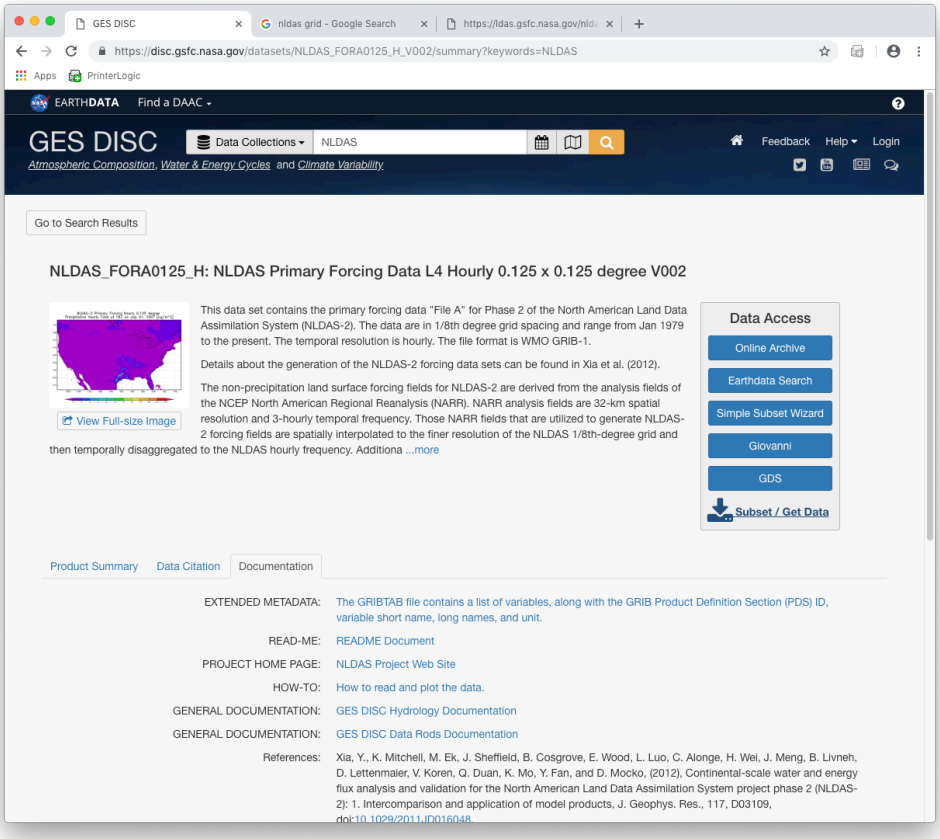


Figure 8.20 NASA GES DISC website for downloading NLDAS-2 data

After registering an Earthdata account on GES DISC website, when you click 'Subset / Get Data' in Fig. 8.20, you will see a pop-up as in Fig. 8.21. After specifying data range and format on this pop-up and pressing 'Get Data' button, you will get a text file that contains download links. Now, you can download NLDAS-2 files on your computer as shown in Fig. 8.22.

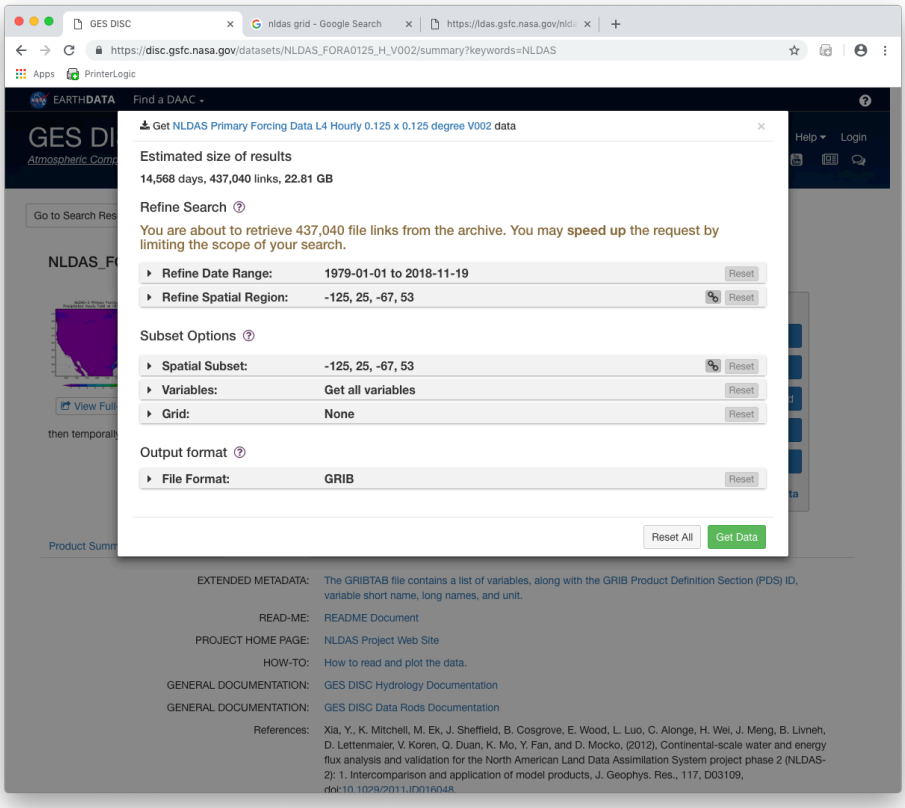


Figure 8.21 Creating text file that contains download link of NLDAS-2 data

Figure 8.22 Shell script to submit que of downloading NLDAS-2 data

For generating the spatial reference file that has NLDAS grid number for each triangle element, a raster file that has NLDAS grid numbers is needed. Fig. 8.23 shows an ascii file (https://ldas.gsfc.nasa.gov/nldas/asc/NLDASmask_UMDunified.asc) that contains grid number and their latitude and longitude of center point. As shown in Fig. 8.24, we can make a point shape file. Furthermore, based on the center point shape file, polygon shape file that resembles the NLDAS-2 grid can be generated as Fig. 8.25. Fig. 8.26 shows the result.

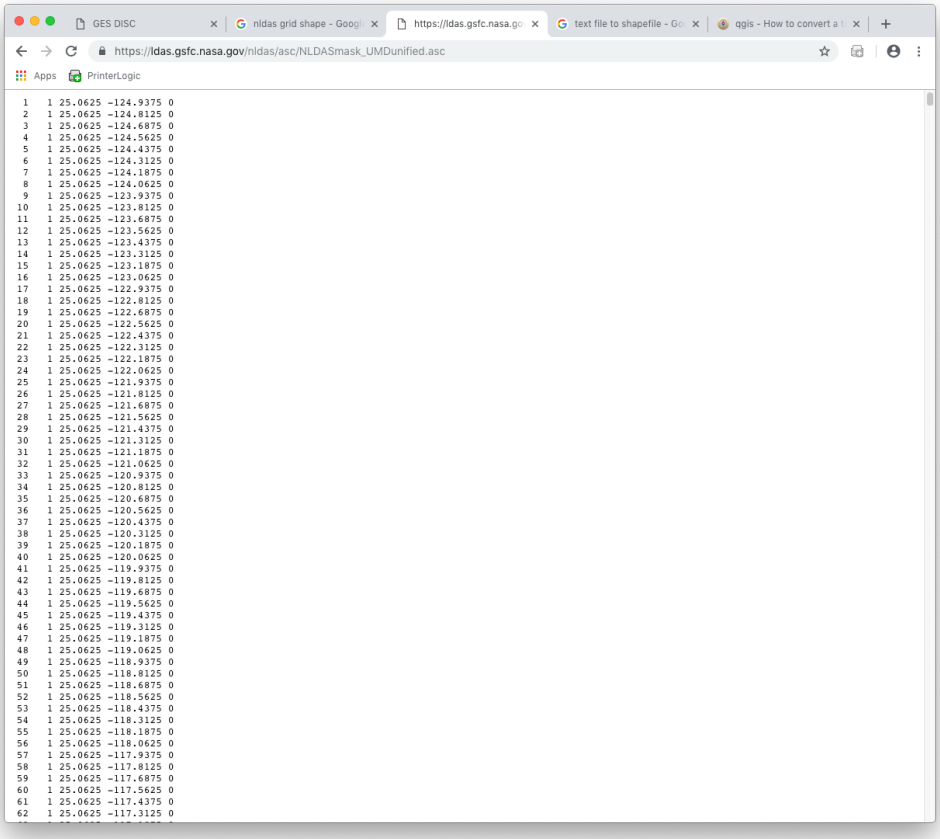


Figure 8.23 The ascii file that contains NLDAS-2 grid numbers and their latitude and longitude of the center point of each grid

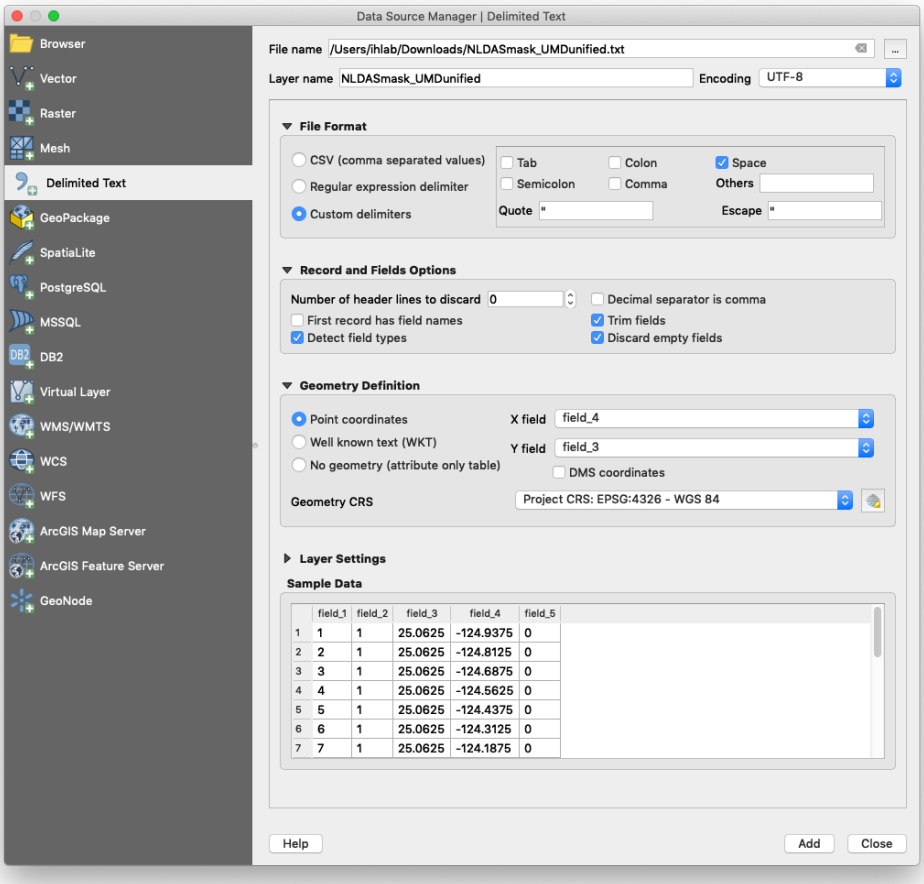
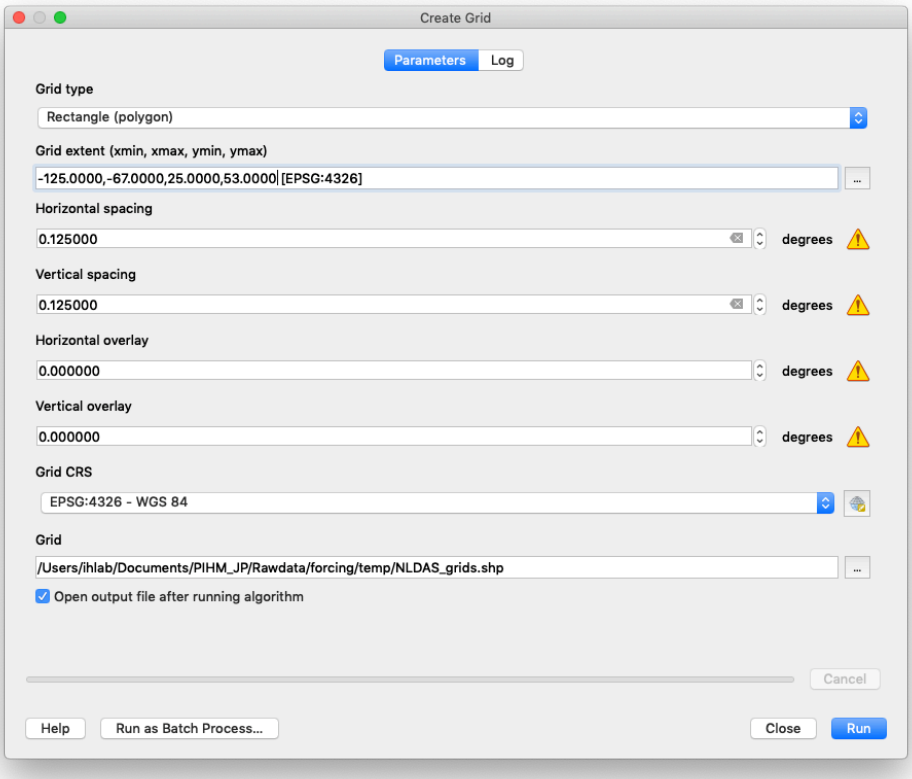
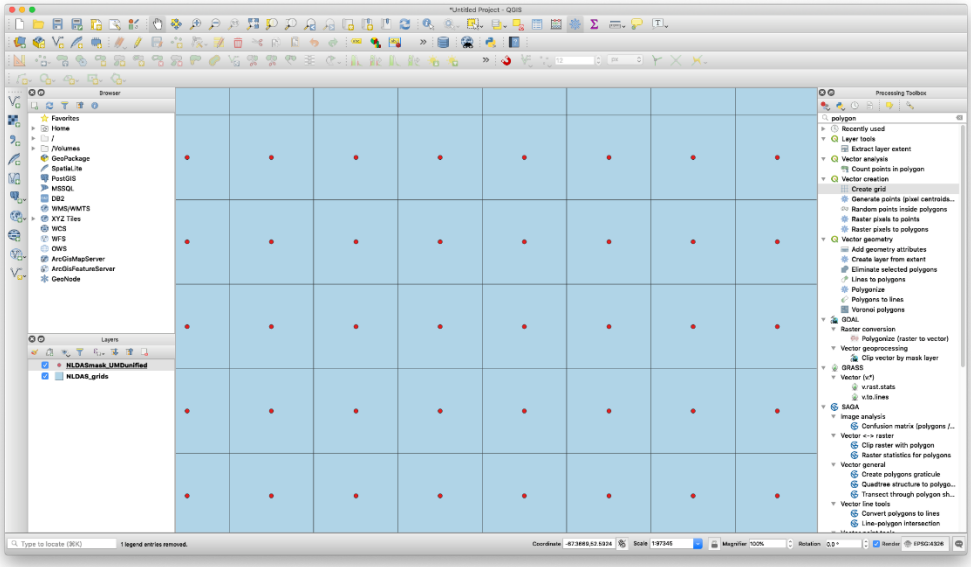


Figure 8.24 The dialog for generating point shape file based on the text of Fig. 8.43



583
584
585 *Figure 8.25 The dialog for generating rectangular polygons that resembles NLDAS-2 grid mesh*

586



587
588
589 *Figure 8.26 Point shape (red circles) and mesh polygons (blue rectangulars) for NLDAS-2 grid*

590

Using Fig. 8.26, we can generate spatial reference of NLDAS-2 grid. Fig. 8.27 shows the buffered target watershed (orange color), selected NLDAS-2 grids that cover the target watershed (yellow), and other grids (blue). You need to export these selected polygons as a shape file and modify attribute table by adding new id number column as shown in Fig. 8.28. This integer will be used when you generate '.forc' file. This polygon shape file should be the '.asc' raster file (This translating procedure is not provided because it was repeated in this tutorial several times).

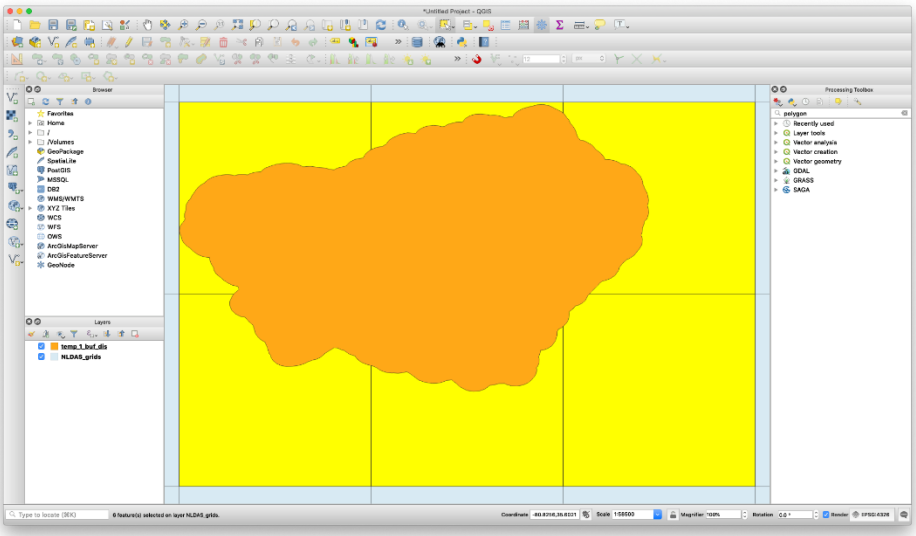
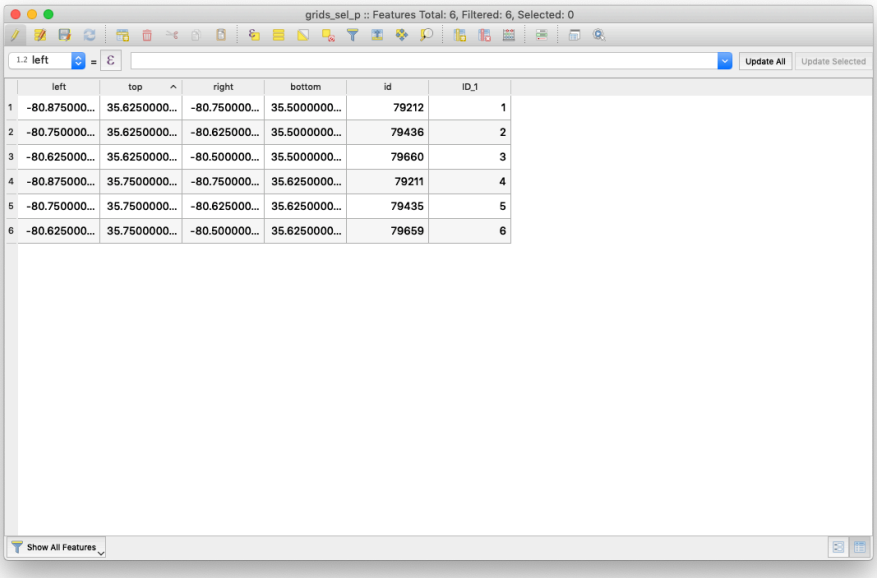


Figure 8.27 Buffered target watershed (orange) and selected NLDAS-2 grids (yellow) to cover target watershed



	left	top	right	bottom	id	ID_1
1	-80.875000...	35.6250000...	-80.750000...	35.5000000...	79212	1
2	-80.750000...	35.6250000...	-80.625000...	35.5000000...	79436	2
3	-80.625000...	35.6250000...	-80.500000...	35.5000000...	79660	3
4	-80.875000...	35.7500000...	-80.750000...	35.6250000...	79211	4
5	-80.750000...	35.7500000...	-80.625000...	35.6250000...	79435	5
6	-80.625000...	35.7500000...	-80.500000...	35.6250000...	79659	6

Figure 8.28 Assign ID numbers for each grid to indicate grids using this ID

9 DataModel Loder

9.1 Introduction

'Data Model Loader' generates input files of PIHM for the mesh generated in 'Domain Decomposition'. PIHM needs 12 files as inputs (mesh, att, riv, soil, geol, lc, init, ibc, para, calib, forc, and projectName.txt).

Some of the input files are generated exactly in the form used by PIHM but others are not. This is because PIHMgis v3.0 was for an older version of PIHM code. In this section, input file format and ways to generate them is detailed.Files generated by 'Data Model Loader' and modified using ancillary codes are expected to generate the input files for PIHM.

9.2 Mesh Data File

Mesh file contains all spatial information of the triangular mesh. This includes number of elements and nodes, neighbor elements for each element, x and y coordinates of node, and elevation of each node are included. The module generate '.mesh' file using previously generated intermediate files. Fig. 9.1 shows the dialog box of this module. Table 9.1 shows the file structure of '.mesh', and Table 9.2 shows description of variables included in the '.mesh' file.

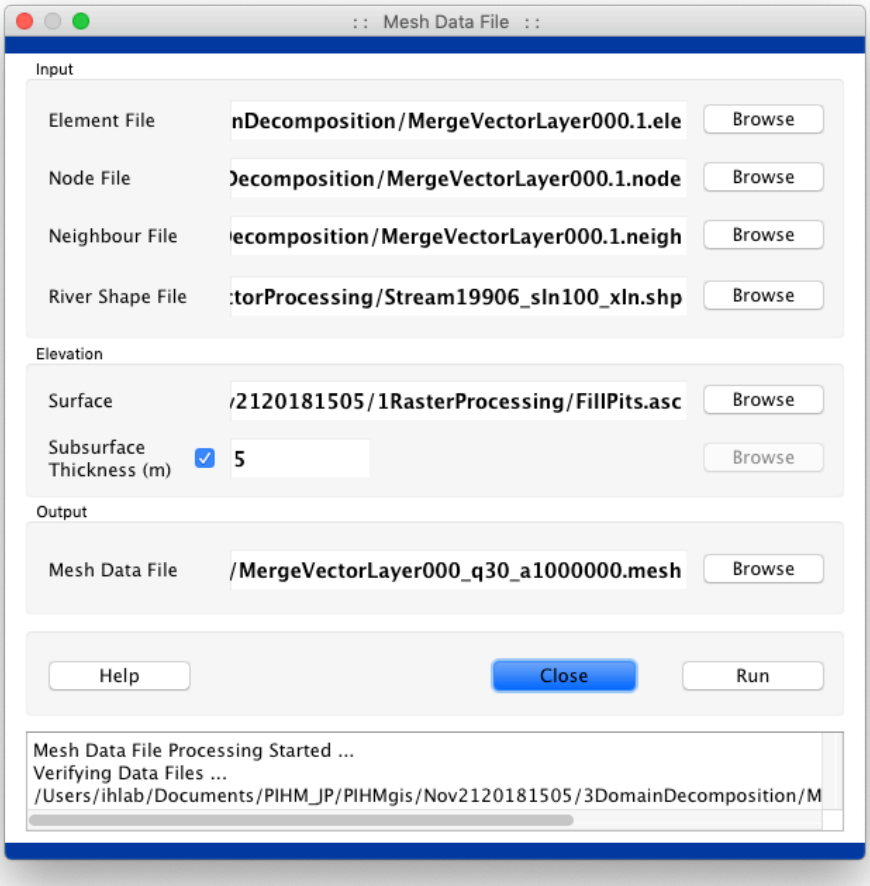


Figure 9.1 Mesh Data File dialog

Table 9.1 File structure of '.mesh'

Numele	NumNode					
Index	Node[0]	Node[1]	Node[2]	Nabr[0]	Nabr[1]	Nabr[2]
Index	Node[0]	Node[1]	Node[2]	Nabr[0]	Nabr[1]	Nabr[2]
…						
…	Repeat NumEle times…					
Index	X	Y	Zmin	Zmax		
Index	X	Y	Zmin	Zmax		
…						
…	Repeat NumNode times…					

Table 9.2 Descriptions of variables in '.mesh'

Variable Name	**Variable Type**	**Variable Description**	**Remarks**
NumEle	Integer	Total number of elements	
NumNode	Integer	Total number of nodes	
Index	Integer	Element index	

Node[0]	Integer	1st node of element	
Node[1]	Integer	2nd node of element	
Node[2]	Integer	3rd node of element	
Nabr[0]	Integer	1st neighbor of element	0: boundary
Nabr[1]	Integer	2nd neighbor of element	0: boundary
Nabr[2]	Integer	3rd neighbor of element	0: boundary
Index	Integer	Node index	
X	Double	X co-ordinate of node	Meters
Y	Double	Y co-ordinate of node	Meters
Zmin	Double	Bed elevation of node	Meters
Zmax	double	Surface elevation of node	Meters

9.3 Att Data File

An '.att' (attribute) file stores class id of physiographic and meteorological properties for each triangular element. If all elements have identical class, that would indicate a homogeneous field within the watershed. In the module, if you choose option 1, it means that one class covers the entire watershed. With option 2, you need to create a lookup table later, and this tutorial follows this option.

Raster layers saved in '.asc' format in section 8 are used here for soil, geology, land cover, and forcing. In absence of geology raster, soil raster is used in its place. Fig. 9.2 and 9.3 show the dialog boxes.

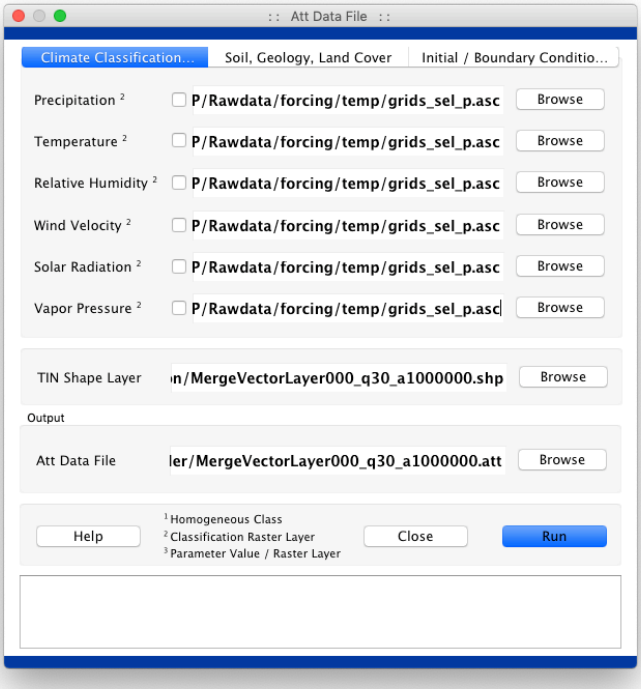


Figure 9.2 Att Data File dialog for 'Climate Classification' tab

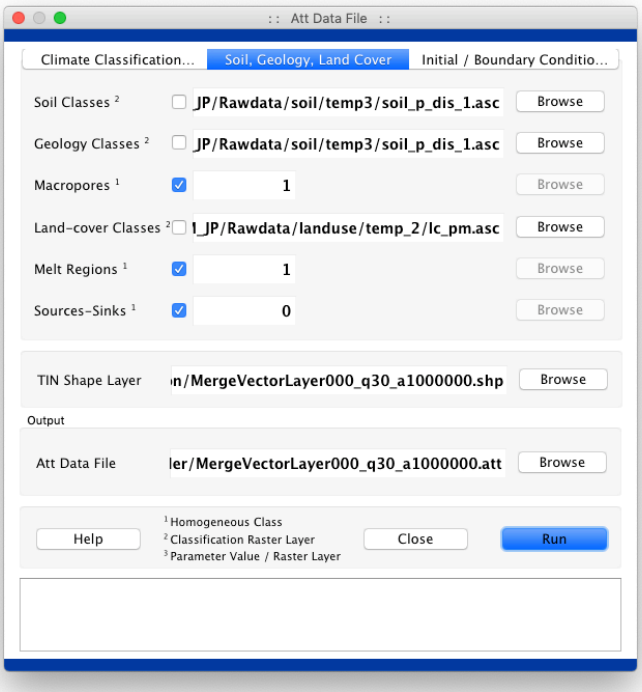


Figure 9.3 Att Data File dialog for 'Soil, Geology, Land Cover' tab

Table 9.3 shows the file structure of '.att', and Table 9.4 shows description of the variables listed in Table 9.3.

Table 9.2 File structure of '.att'

Index	Soil	Geol	LC	IS_LC	Snw_IC	Sfc_IC	Ust_IC	St_IC	Ppt	Tmp	Rh	Wnd	Rn	G	…
Index	Soil	Geol	LC	IS_LC	Snw_IC	Sfc_IC	Ust_IC	St_IC	Ppt	Tmp	Rh	Wnd	Rn	G	…
…															…
…					Repeat NumEle times…										…

…	G	VP	S	mF	BC[0]	BC[1]	BC[2]	mP
…	G	VP	S	mF	BC[0]	BC[1]	BC[2]	mP
…								
…				Repeat NumEle times…				

Table 9.4 Descriptions for variables in '.att'

Variable Name	Variable Type	Variable Description	Remarks
Index	Integer	Element index	
Soil	Integer	Soil class	
Geol	Integer	Geology class	
LC	Integer	Land cover class	
IS_IC	Integer	Interception storage	Initial condition
Snw_IC	Double	Snow accumulation	Initial condition
Srf_IC	Double	Surfaceflow state	Initial condition
Ust_IC	Double	Unsaturated state	Initial condition
St_IC	Double	Saturated state	Initial condition
Ppt	Integer	Precipitation series	
Tmp	Integer	Temperature series	
RH	Integer	Relative humidity series	
Wnd	Integer	Wind velocity series	
Rn	Integer	Solar radiation series	
G	Integer	Dummy	
VP	Integer	Vapor pressure	
S	Integer	Source/sink	
mF	Integer	Melt factor series	
BC[0]	Integer	Boundary condition type on edge	0 is No flow; 1 is head; and 2 is flux boundary cond.
BC[1]	Integer	…	
BC[2]	Integer	…	
mP	Integer	Macropore present or not	1: Yes / 0: No

9.4 Riv Data File

Topological information related to river segments is stored in this file. Fig. 9.4 shows the dialog box for generating this file in PIHMgis, and Table 9.5 and 9.6 shows the structure and descriptions of variables in the generated '.riv' file.

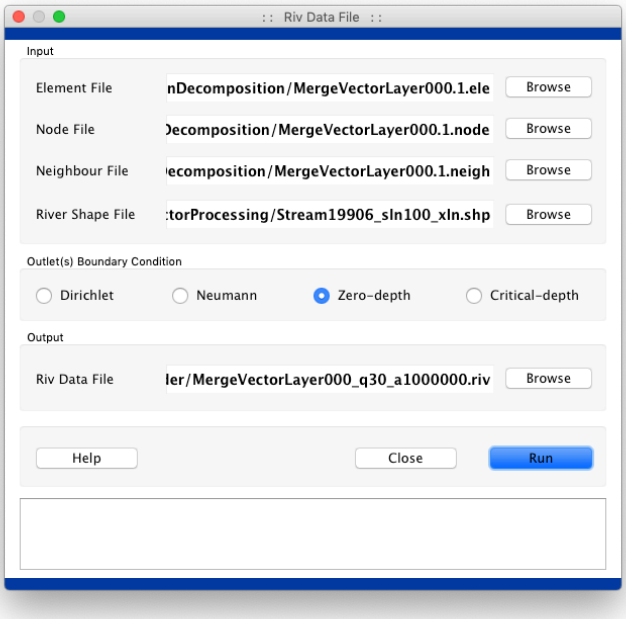


Figure 9.4 Riv Data File dialog

673 *Table 9.5 File structure of '.riv'*

NumRiv										
Index	FromNode	ToNode	Down	LeftEle	RightEle	Shape	Material	IC	BC	Res
…										
…		Repeat NumRiv times…								
“Shape”	Numshape									
Index	Depth	InterpOrd	WidCoeff							
Index	Depth	InterpOrd	WidCoeff							
…		Repeat NumShape times…								
“Material”	NumMat									
Index	n	Cwr	KsatH	KsatV	Bed					
Index	n	Cwr	KsatH	KsatV	Bed					
…		Repeat NumMat times…								
“IC”	NumIC									
Index	Value									
Index	Value									
...		Repeat NumIC times…								
“BC”	NumBC									
Type	Index	Length								
Time	Value									
…		Repeat Length times…								
Type	Index	Length								
…		Repeat NumBC times…								
“Res”	NumRes									

674

675 *Table 9.6 Description of variables in '.riv'*

Variable Name	Variable Type	Variable Description	Remarks
NumRiv	Integer	Number of river segments	
Index	Integer	River segment ID	Begin with 1
FromNode	Integer	From node ID	
ToNode	Integer	To node ID	
Down	Integer	Downstream segment ID	
LeftEle	Integer	Left element ID	
RightEle	Integer	Right element ID	
Shape	Integer	Shape ID	
Material	Integer	Material ID	
IC	Integer	Initial condition ID	
BC	Integer	Boundary condition ID	
Res	Integer	Reservoir ID	
Numshape	Integer	Number of shape types	
Index	Integer	Shape ID	Begin with 1
Dummy	-	-	
Depth	Double	Depth of river segment	
InterpOrder	Integer	Interpolation order *	1 if a rect.
WidCoeff	Double	Width coefficient *	Width if a rect.
NumMat	Integer	Number of material types	
Index	Integer	Material ID	Begin with 1
n	Double	Manning's roughness coeff.	
Cwr	Double	Discharge coefficient	0.6
KsatH	Double	Size hydraulic conductivity	
KsatV	Double	Bed hydraulic conductivity	
Bed	Double	Bed depth	
NumIC	Integer	Number of init. cond. types	
Index	Integer	Initial condition ID	
Value	Double	Initial condition water table	
NumBC	Integer	Number of boundary cond.	
Type	Integer	Boundary condition type	
Index	Integer	Boundary condition ID	
Length	Integer	Length of BC TimeSeries	
Time	Double	Time	(days)
Value	Double	BC value	(m or m/day)
NumRes	Integer	Number of reservoirs	
* Interpolation order (b) and width coefficient (a) are parameters defining relation between width and depth of a river segment as: [$D=ax(W/2)^b$]			

676

9.5 Soil Data File

Soil module uses pedo-transfer functions (Wösten, J.H.M. et al., 2001) to estimate soil hydraulic parameters from soil texture data. Soil hydraulic parameters required by the PIHM are: vertical soil hydraulic conductivity, porosity, residual porosity, macropore depth, macropore vertical hydraulic conductivity, van-Genuchten drainage (alpha and beta) parameters. For using pedo-transfer functions, following soil textual properties are needed: (1) silt percentage, (2) clay percentage, (3) organic matter (optional), and (4) bulk density (optional). Fig. 9.5 shows the dialog of soil module. Fig. 9.6 shows the soil input text file for this module. This text file is based on the extracted soil properties from SSURGO database that you've obtained in section 8.2. The file should look like as shown in Fig. 9.6. Table 9.7 shows the format of the input text file shown in Fig. 9.6. Table 9.8 shows the file structure of '.soil', which is the output of this module and input to the PIHM model. Table 9.9 shows description of variables in '.soil'.

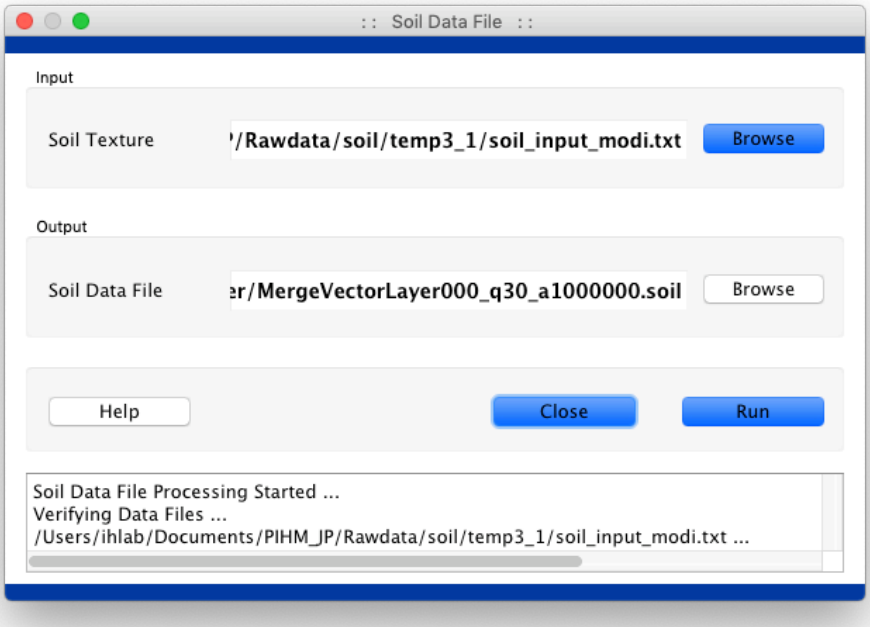


Figure 9.5 Soil Data File dialog

soil_input_modi.txt

ID	SILT	CLAY	OM	BD
1	17.30	33.50	0.32	1.51
2	32.90	38.30	0.63	1.39
3	23.90	36.70	0.28	1.53
4	14.70	33.20	0.30	1.38
5	22.70	20.90	0.83	1.46
6	24.30	31.70	0.30	1.46
7	24.30	32.80	0.30	1.46
8	32.20	28.00	0.35	1.47
9	32.20	28.00	0.35	1.47
10	29.60	26.60	0.35	1.41
11	29.60	26.60	0.35	1.41
12	32.20	28.00	0.35	1.47
13	22.70	28.80	0.28	1.44
14	33.50	34.80	0.32	1.41
15	23.00	30.20	0.35	1.46
16	24.80	30.20	0.35	1.46
17	16.80	26.20	0.35	1.35
18	16.80	26.20	0.35	1.35
19	16.80	26.20	0.35	1.35
20	16.80	26.20	0.35	1.35
21	19.10	16.60	0.35	1.42
22	19.10	16.60	0.35	1.42
23	38.50	19.70	1.03	1.35
24	20.80	30.20	0.52	1.40
25	20.80	30.20	0.52	1.40
26	31.00	34.00	0.30	1.41
27	14.00	30.00	0.50	1.60
28	14.00	30.00	0.50	1.60
29	38.80	35.40	0.52	1.36

Figure 9.6 Soil input file based on extracted parameters in section 8.2

Table 9.7 Format of input text file including soil textual properties for Soil Data File module

One line description about the soil texture file				
MUKEY 1	SILT %	CLAY %	ORGANIC MATTER	BULK DENSITY
MUKEY 2	SILT %	CLAY %	ORGANIC MATTER	BULK DENSITY
…	…	…	…	…
…	…	…	…	…
MUKEY n	SILT %	CLAY %	ORGANIC MATTER	BULK DENSITY

Table 9.8 File structure of '.soil'

NumSoil								
Index	KsatV	ThetaS	ThetaR	infD	Alpha	Beta	hAreaF	macKsatV
Index	KsatV	ThetaS	ThetaR	infD	Alpha	Beta	hAreaF	macKsatV
…								
…	Repeat NumSoil times…							

706 *Table 9.9 Description of variables in '.soil'*

Variable Name	Variable Type	Variable Description	Remarks
NumSoil	Integer	Number of soil classes	
Index	Integer	Soil class number	Begin with 1
KsatV*	Double	Vertical saturated hydraulic conductivity	
ThetaS	Double	Porosity	
ThetaR	Double	Residual porosity	
infD	Double	Top soil layer across which infiltration is calculated	Generally set to 0.1 m
Alpha*	Double	Van Genuchten soil parameter	
Beta*	Double	Van Genuchten soil parameter	
hAreaF	Double	Horizontal area fraction of macropore	
macKsatV	Double	Vertical macropore hydraulic conductivity	
* Note: Beta in $K_u - S$ Van Genuchten relationship is used as $K_u = S^{0.5}\left(1-\left(1-S^{\frac{\beta}{\beta-1}}\right)^{\frac{\beta-1}{\beta}}\right)^2$			

707

9.6 Geol Data File

If you have any form of measured data regarding hydraulic properties of the geology, you can use 'Geol Data File' module to generate one. This module is quite similar with 'Soil Data File' module you worked on in section 9.5. It also uses pedo-transfer function (Wösten, J.H.M. et al., 2001) for generating several parameters. You can use exactly same text file that you used in 'Soil Data File', if you do not have geology information. Fig. 9.6, as the input text file for 'Geol Data File' module. Fig. 9.7 shows the dialog of this module. Table 9.10 shows the file structure of '.geol', which is the result of this module. Table 9.11 shows the description of variables in '.geol'.

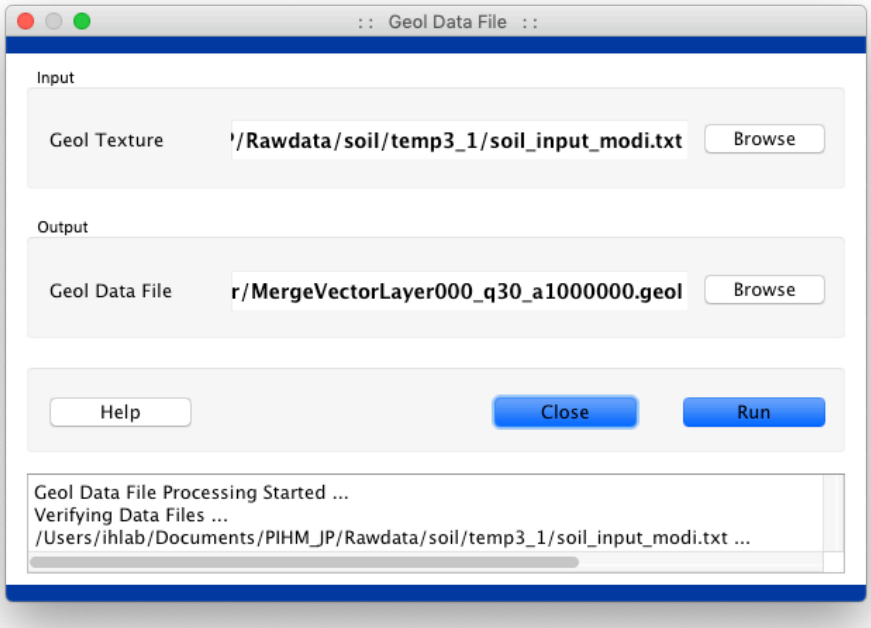


Figure 9.7 Geol Data File dialog

Table 9.10 File structure of '.geol'

NumGeol										
Index	KsatH	KsatV	ThetaS	ThetaR	infD	Alpha	Beta	vAreaF	macKsatH	macD
Index	KsatH	KsatV	ThetaS	ThetaR	infD	Alpha	Beta	vAreaF	macKsatH	macD
…										
…	Repeat NumGeol times…									

Table 9.11 Description of variables in '.geol'

Variable Name	Variable Type	Variable Description	Remarks

NumSoil	Integer	Number of soil classes	
Index	Integer	Soil class number	Begin with 1
KsatV	Double	Vertical saturated hydraulic conductivity	
ThetaS	Double	Porosity	
ThetaR	Double	Residual porosity	
infD	Double	Top soil layer across which infiltration is calculated	Generally set to 0.1 m
Alpha	Double	Van Genuchten soil parameter	
Beta	Double	Van Genuchten soil parameter	
vAreaF	Double	Vertical area fraction of macropore	
macKsatH	Double	Horizontal macropore hydraulic conductivity	
macD	Double	Macropore depth	

9.7 LC Data File

As mentioned in section 8.3, this module in PIHMgis had been designed assuming NLCD 2001 classification as the base data. Therefore, it simply requires NLCD land cover types in ascending order in a text file, as shown in Fig. 9.9.

Since we are using NLDAS data in this use case, which uses UMD Land Cover Classification (UMDLCC), we have reclassified the UMDLCC to make it consistent with NLCD 2001 classification. The module is run as shown in Fig. 9.8. Table 9.12 shows the input file format. Table 9.13 shows the file structure of '.lc'. Table 9.14 shows the description of variables in '.lc'.

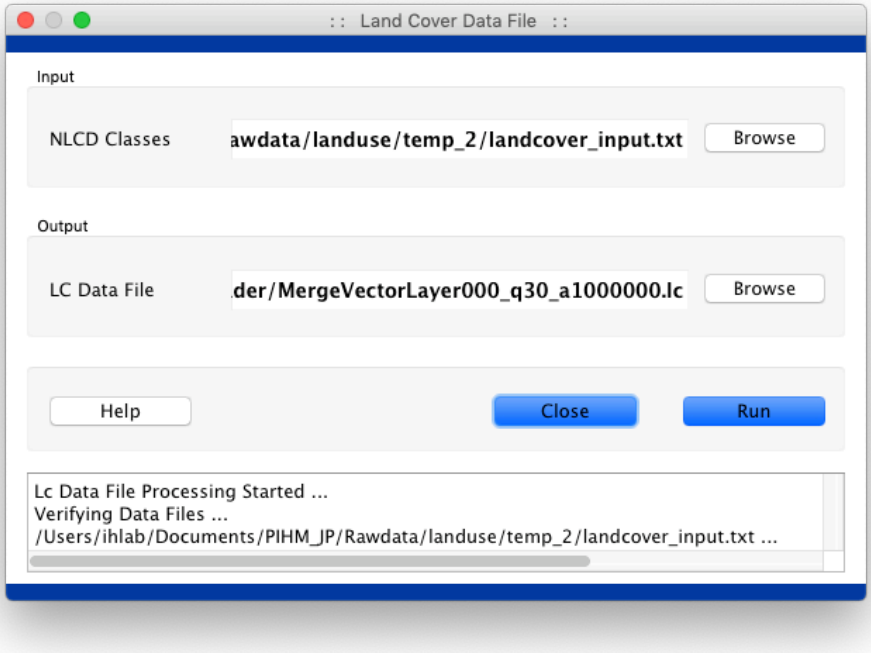


Figure 9.8 Land Cover Data File dialog

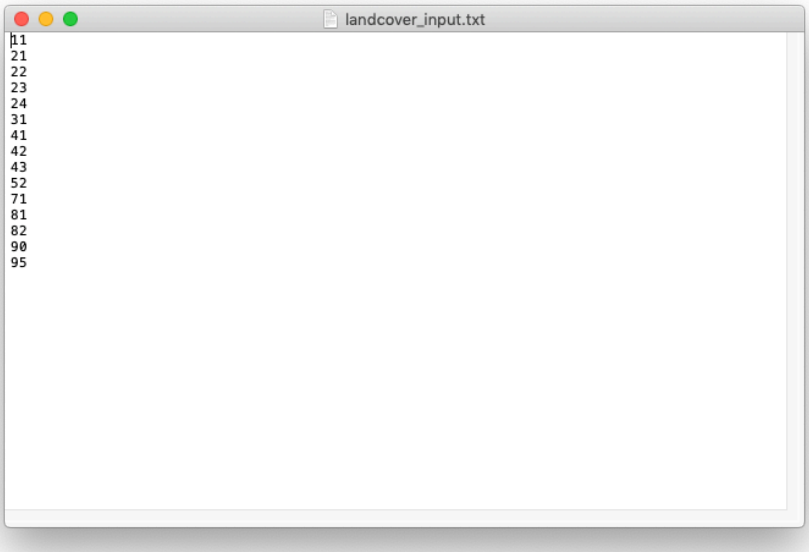


Figure 9.9 Land cover input file based on the NLCD 2011 legend numbers

Table 9.12 Format of input text file for Land Cover Data File module

Class_1
…
…
Class_n

Table 9.13 File structure of '.lc'

NumLC							
Index	LAIamx	Rmin	RS_ref	Albedo	VegFrac	n	RzD
Index	LAImax	Rmin	RS_ref	Albedo	VegFrac	n	RzD
…							
…		Repeat NumLC times…					
…							
Index	LAImax	Rmin	RS_ref	Albedo	VegFrac	n	RzD

Table 9.14 Description of variables in '.lc'

Variable Name	**Variable Type**	**Variable Description**	**Remarks**
NumLC	Integer	Number of land cover classes	
Index	Integer	Land cover class number	
LAImax	Double	Maximum LAI	
Rmin	Double	Minimum stomatal resistance	
Rs_ref	Double	Reference stomatal resistance	
Albedo	Double	Albedo	
VegFrac	Double	Vegetation fraction	
n	Double	Manning's roughness coeff.	Day/$m^{1/3}$

RzD	Double	Root zone depth	

9.8 Init Data File

Initial file ('.init') has initial state for all the hydrologic variables. This module facilitates a uniform initialization of individual state variable for the domain. However, this file is an optional file. If you have some observation data with which you would like to initialize the model, you should use the interpolated raster data in the '.att' file. Otherwise, you can write specific values that you obtained from the results of spin up run. Fig. 9.10 shows the dialog of this module. You can set head depths of interception, snow, and surface. Also, unsaturated zone (shown as Soil Moisture), saturated zone (shown as Groundwater), river, and riverbed can be set by exact depth in meter or percent based on domain depth inserted earlier.

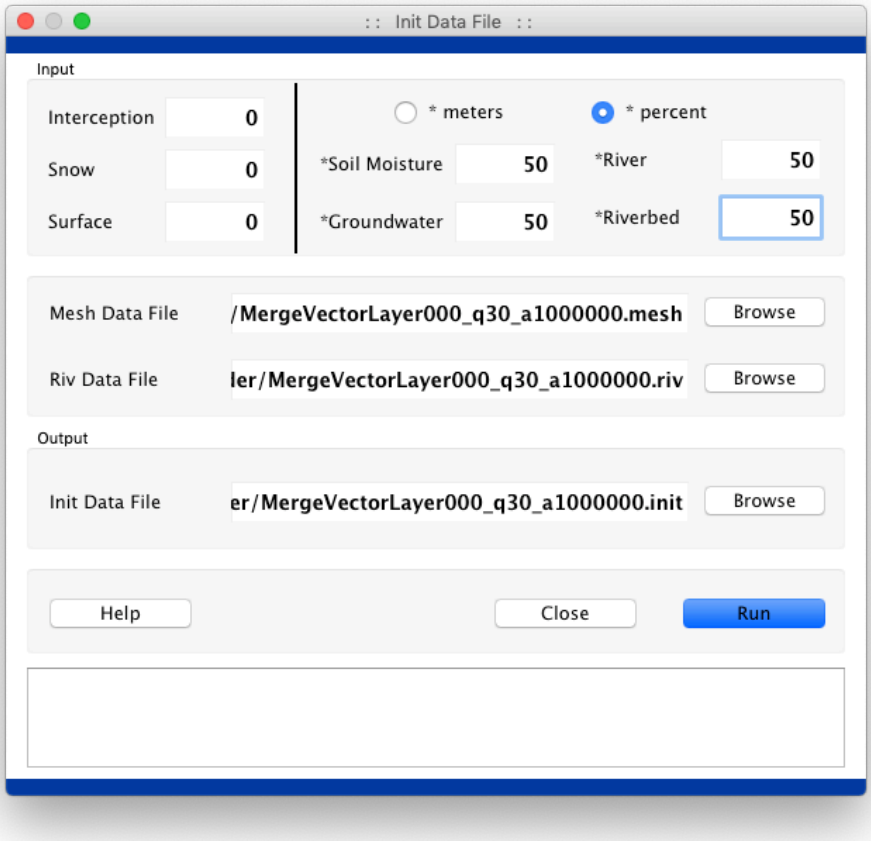


Figure 9.10 Init Data File dialog

However, you should know that '.init' file generated by PIHMgis v3.0 should be modified for using in current version of PIHM. It's because an unsaturated zone in groundwater layer is added in the model for this current version of PIHM. The '.init' file has 5

columns for each elements when it generated from PIHMgis v3.0. However, now 6 columns are needed for the current version of PIHM. As shown in the Table 9.15, 5 columns are for interception, snow, overland, unsat, and sat. But PIHMgis v3.0 has 3 layer (surface zone, unsaturated zone, and groundwater zone), and the depth of unsaturated region in groundwater zone should be provided. Current version of PIHM reads this information from sixth column of '.init' file.

Table 9.15 File structure of '.init'

IS	Snow	Overland	UnSat	Sat	UnSat2*
IS	Snow	Overland	UnSat	Sat	UnSat2*
Repeat NumEle times…					
RiverState	Sat Beneath River				
RiverState	Sat Beneath River				
Repeat NumRiv times…					
* If there is no sixth column for UnSat2, the depth of second unsaturated layer, you should create the column values. In absence of relevant data, use values such that the saturation in UnSat2 is the same as in the top unsaturated zone.					

9.9 IBC Data File

This module generates '.ibc' that contains any boundary condition information if present for the modeling domain. PIHM has provision to specify both Dirichlet and Neumann boundary conditions. The scope of this module is limited to just creating a dummy file, which is generally the case for a modeling domain. It generates a dummy file, which allows setting up a simulation. If you have any information for this file, you can modify the 'ibc' file based on Table 9.16 and 9.17.

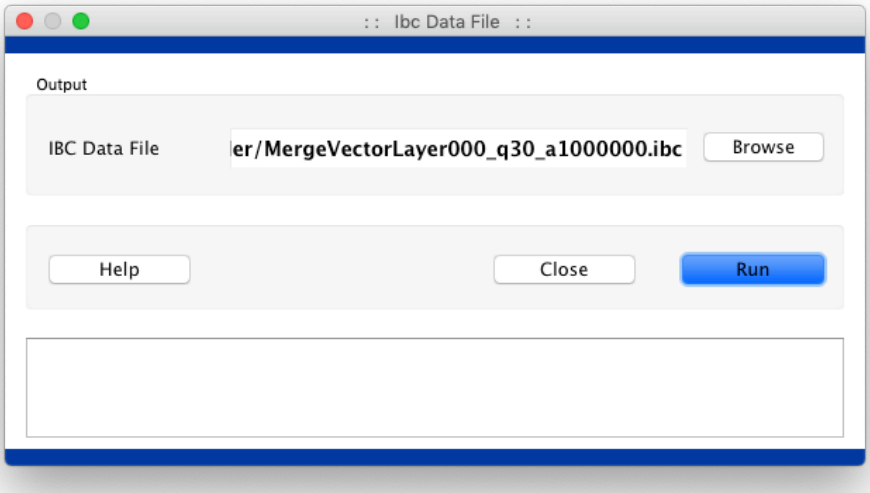


Figure 9.11 Ibc Data File dialog

Table 9.16 File structure of '.ibc'

NumBC1	NumBC2	
"BC1"	Index	Length
Time	Value	
Repeat Length times…		
Time	Value	
"BC1"	Index	Length
Repeat NumBC1 times…		
"BC1"	Index	Length
"BC2"	Index	Length
Time	Value	
Repeat Length times…		
Time	Value	
"BC2"	Index	Length
Repeat NumBC2 times…		
"BC2"	Index	Length

Table 9.17 Description of variables in '.ibc'

Variable Name	**Variable Type**	**Variable Description**	**Remarks**
NumBC1	Integer	Number of Dirichlet BC	
NumBC2	Integer	Number of Neumann BC	
Index	Integer	Boundary condition ID	
Length	Integer	Number of time steps	
Time	Double	Time	
Value	Double	Value	(m or m/day)

9.10 Param Data File

Parameter file provides all the control information for running PIHM. It contains solver options, model modes, and parameters that govern model error. You can see the several model and control panel from the Table 9.18 and Table 9.19. With that, you can choose appropriate models and parameters that are provided on the dialog as shown in Fig. 9.12. Fig. 9.12 only shows 'Model Output Parameters' tab, but you can also set your configurations discussed above at 'Model Process Parameters' tab and 'Model Solver Parameters' tab.

Notably, '.para' file from PIHMgis v3.0 is no longer valid for the current version of the PIHM. After generating '.para' file from PIHMgis v3.0, you should modify this file to make it consistent with Table 9.18 and Table 9.19 (or as a starter, use the example file that comes with the code).

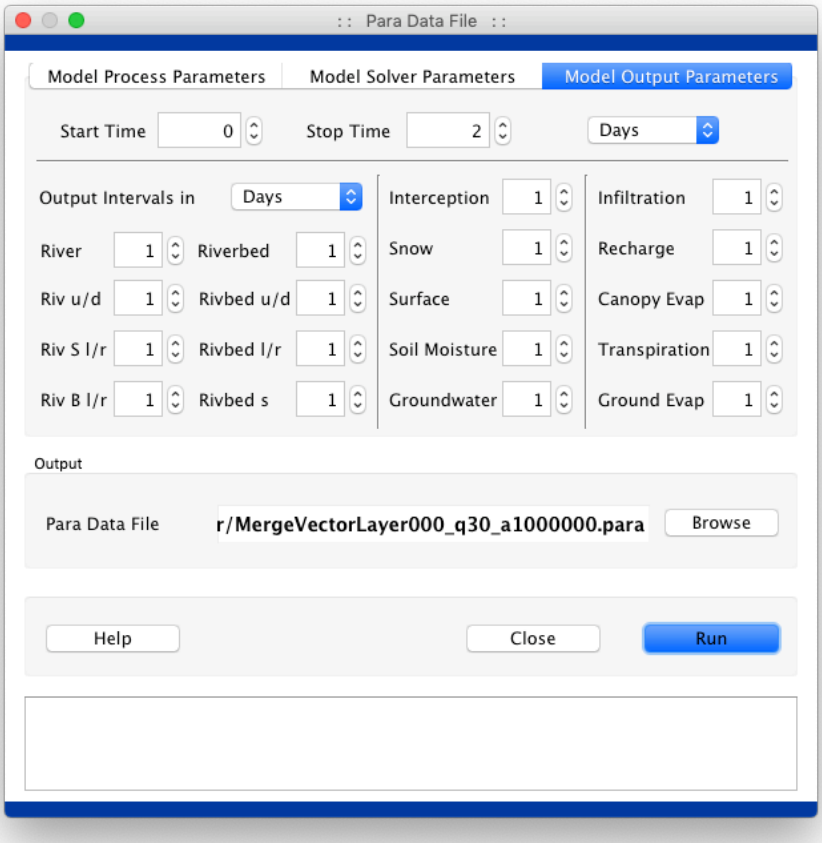


Figure 9.12 Para Data File dialog

814 *Table 9.18 File structure of '.param'*

Verbose	Debug	Init_type	totFiles			
PNetP	PIS	Psurf	Punsat	PGW		
Pet0	Pet1	Pet2	PP	PRech		
PunsatT	PRechT	Pstage	Psubriv			
PrivFlx0	PrivFlx1	PrivFlx2	PrivFlx3	PrivFlx4	…	PrivFlx10
NetPInt	ISInt	surfInt	unsatInt	GWInt		
et0Int	et1Int	et2Int	PInt	RechInt		
unsatTInt	RechTInt	stageInt	subrivInt			
rivFlxInt	…					
SatMode	RivMode					
Solver	GSType	MaxK	Delta			
AbsTol	RelTol	InitStep	MaxStep	ETstep		
StartTime	EndTime	Output				
a	b					

815
816 *Table 9.19 Description of variables in '.param'*

Variable Name	**Variable Type**	**Variable Description**	**Remarks**
Verbose	Integer	Verbose mode	Yes/no: 1/0
Debug	Integer	Debug mode	Yes/no: 1/0
Init_type	Integer	State initialization type	Relax(0), AttFile(1), InitFile(3)
totFiles	Integer	Number of output files	25 suggested
PNetP, PIS, Psurf, Punsat, PGW, Pet0, Pet1, Pet2, PP, PRech, PunsatT, PRechT, Pstage, Psubriv	Integer	Print: Net precipitation, Interception Storage, Surface Water, Unsaturated Storage, Groundwater, Interception Loss, Transpiration, Evaporation from Ground, Precipitation, Recharge to Groundwater, Unsaturated Storage in Groundwater Layer, Recharge to Unsaturated zone in Groundwater Layer, River Stage, Sub River Stage	Yes/no: 1/0
PrivFlx0, PrivFlx1	Integer	Print: Longitudinal {Flow to, Flow from} a river element	Yes/no: 1/0
PrivFlx2, PrivFlx3	Integer	Print: Lateral Overland Flow to a river element from {left, right}	Yes/no: 1/0
PrivFlx4, PrivFlx5	Integer	Print: Lateral Groundwater Flow to a river element from {left, right}	Yes/no: 1/0

PrivFlx6	Integer	Print: Leakage/Base Flow to/from aquifer	Yes/no: 1/0
PrivFlx7, PrivFlx8	Integer	Print: Longitudinal {Flow to, Flow from} a aquifer element beneath river	Yes/no: 1/0
PrivFlx9, PrivFlx10	Integer	Print: Lateral Groundwater Flow to an aquifer element from {left, right} beneath river	Yes/no: 1/0
NetPInt, ISInt, surfInt, unsatInt, GWInt, et0Int, et1Int, et2Int, PInt, RechInt, unsatTInt, RechTInt, stageInt, subrivInt, rivFlxInt	Integer	Print Interval: Net precipitation, Interception Storage, Surface Water, Unsaturated Storage, Groundwater, Interception Loss, Transpiration, Evaporation from Ground, Precipitation, Recharge to Groundwater, Unsaturated Storage in Groundwater Layer, Recharge to Unsaturated zone in Groundwater Layer, River Stage, Sub River Stage, River Flow	Note: Unit is in minutes
SatMode	Integer	Saturation formulation	Kinematic(1), Diffusion(2)
RivMode	Integer	River formulation	Kinematic(1), Diffusion(2)
Solver	Integer	Cvode Solver Type	Iterative(2)
GSType	Integer	GS Solver Type	Kinematic(1), Diffusion(2)
MaxK	Integer	Max Krylov dimension	
Delta	Double	GMRES convergence criterion	
AbsTol	Double	Absolute Tolerance	
RelTol	Double	Relative Tolerance	
InitStep	Double	Initial time-step	[see SUNDIALS manual]
MaxStep	Double	Maximum time-step	[see SUNDIALS manual]
Etstep	Double	ET time-step	
StartTime	Double	Simulation start time	
EndTime	Double	Simulation end time	
Output	Double	Output step-size	
a *	Double	Step-size factor	
b *	Double	Base step-size	
* stepsize = $b \times a^i$			

817
818

9.11 Calib Data File

As the name suggests, `.calib' file is used for calibration. Several physical parameters have been identified for the purpose of calibration. The values you choose in the module for specific parameter is a multiplier hence it globally and uniformly nudges the a-priori parameter value across the whole modeling domain.

On the dialog as shown in Fig. 9.13, you will notice a default value has been provided for most of the parameters. A value 1.0 implies that it would not change the parameter (since it is a multiplier to any parameter). You can accept these default values for the purpose of first run and come back later if you decide to change any of them.

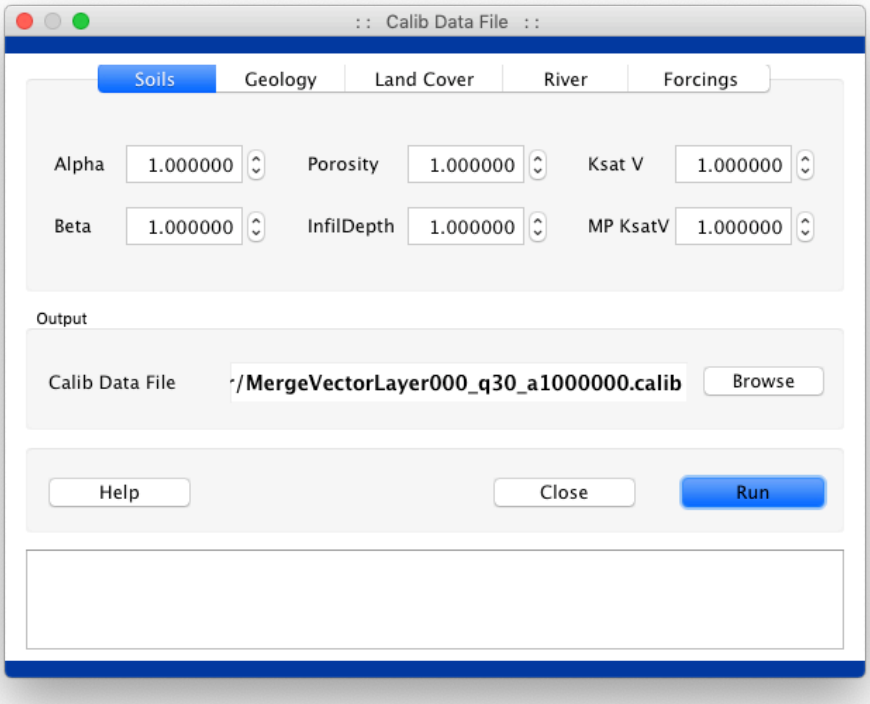


Figure 9.13 Calib Data File dialog

Table 9.20 File structure of '.calib'

geolKsatH	geolKsatV	soilKsatV	macKsatH	macKsatV
infD	RzD	macD		
Porosity	Alpha	Beta		
vAreaF	hAreaf			
VegFrac	Albedo	Rough		
Precep	Temp			
Et0	Et1	Et2		
rivRough	rivKsatH	rivKsatV	rivBedThickness	
rivDepth	rivWidCoeff			

Table 9.21 Description of variables in '.calib'

Variable Name	**Variable Type**	**Variable Description**	**Remarks**
infD	Double	Infiltration depth	
RzD	Double	Root zone depth	
macD	Double	Macropore depth	
Alpha, Beta	Double	Van genuchten parameters	
vAreaF, hAreaF	Double	Vertical, horizontal macropore area function	
VegFrac	Double	Vegetation Fraction	
Albedo	Double	Albedo	
Rough	Double	Manning's n	
Precep	Double	Precipitation	
Temp	Double	Temperature	
Et0	Double	Interception Loss	
Et1	Double	Transpiration	
Et2	Double	Evaporation from Ground	
rivRough	Double	River Manning's n	
rivKsatH, rivKsatV	Double	Conductivity of river walls and bed	
rivBedThickness	Double	River bed thickness	
rivDepth	Double	River depth	
rivWidCoeff	Double	River width coefficient	

9.12 Forc Data File

There is no module for generating forcing file in PIHMgis. This tutorial provides detail regarding how to generate forcing file with Table 9.22 and 9.23 and captures of sample forcing file from Fig. 9.14 to 9.17. Sample forcing file uses NLDAS-2 data [10] of one year (1.1.2017-

[10] NLDAS Primary Forcing Data L4 hourly 0.125 x 0.125 degree V002 (https://disc.gsfc.nasa.gov/datasets/NLDAS_FORA0125_H_V002/summary)

12.31.2017) for six grids (354,85-356,86), first discussed in section 3.2 and downloaded in section 8.4. Program codes used in this tutorial for extracting data from NLDAS-2 grib files and generating '.forc' file from such data are detailed in the Appendix A.

At first, you should know that 'Index' used from "Prep" to "VP" is the id numbers you generated in section 8.4. The '.asc' raster file was used in section 9.3 to generate '.att' file. Time in the .forc time series is recorded in daily units, however you may represent as fine of a data as needed.

For "LAI" and "RL", the 'Index' follows the id number of land cover types generated in section 8.3. These time series can be obtained from remote sensing time-series layers by averaging values on each land cover types repeatedly. Otherwise, you can use the table of NLDAS vegetation parameters[11] as this tutorial did. It provides averaged LAI values and RL values for each UMD LC class for each month. With this table, you can write like Fig. 9.16. IsFactor was assumed as 0.0002 for all land cover types.

Finally, this tutorial assumed there is no snow in the target area, so MF is set to 0 as shown in Fig. 9.17. Furthermore, this tutorial also assumed there is no source and sink, so NumSS is assumed as 0 in Fig. 9.14.

[11] NLDAS Vegetation Parameters (https://ldas.gsfc.nasa.gov/nldas/NLDASmapveg.php)

Table 9.22 File struvture of '.forc'

NumPrep	NumTemp	NumRH	NumWind	NumRN	NumG	NumVP	NumLC	NumMF	NumSS
"Prep"	Index	Length							
Time	Value								
Repeat Length times…									
Time	Value								
"Prep"	Index	Length							
Repeat NumPrep times…									
"Temp"*	Index	Length							
"RH"*	Index	Length							
"Wind"*	Index	Length	Height						
"Rn"*	Index	Length							
"VP"*	Index	Length							
"LAI"*	Index	Length	IsFactor						
"RL"*	Index	Length							
"MF"*	Index	Length							
"SS"*	Index	Length							
* Same as "Prep" times-series									

Table 9.23 Description of variables in '.forc'

Variable Name	Variable Type	Variable Description	Remarks
NumPrep	Integer	Number of precipitation time-series	
NumTemp	Integer	Number of temperature time-series	
NumRH	Integer	Number of relative humidity time-series	
NumWind	Integer	Number of wind velocity time-series	
NumRn	Integer	Number of solar radiation time-series	
NumG	-	Dummy	
NumVP	Integer	Number of vapor pressure time-series	
NumLC	Integer	Number of land cover types for LAI and rough length time-series	
NunMF	Integer	Number of melt factor time-series	
NumSS	Integer	Number of source/sink	
Index	Integer	Time-series ID	Index follows NumVar specified in 1st line
Length	Integer	Number of time steps	
Time	Double	Time	
Value	Double	Data value	
Height	Double	Height of wind velocity observation	
IsFactor	Double	Maximum Interception storage factor	

```
1   6    6    6    6    6    0    6    14    1    0
2   PP   1   17520
3   0.000000 0.000000
4   0.041660 0.000000
5   0.041670 0.000000
6   0.083320 0.000000
7   0.083330 0.000000
8   0.124990 0.000000
9   0.125000 0.000000
10  0.166660 0.000000
11  0.166670 0.000000
12  0.208320 0.000000
13  0.208330 0.000000
14  0.249990 0.000000
15  0.250000 0.000000
16  0.291660 0.000000
17  0.291670 0.000000
18  0.333320 0.000000
19  0.333330 0.000000
20  0.374990 0.000000
21  0.375000 0.003830
22  0.416660 0.003830
23  0.416670 0.012250
24  0.458320 0.012250
25  0.458330 0.010176
26  0.499990 0.010176
27  0.500000 0.026035
28  0.541660 0.026035
29  0.541670 0.010387
30  0.583320 0.010387
31  0.583330 0.010733
```

Figure 9.14 Screenshot of '.forc': header line and precipitation data

```
17518   364.916656 0.000000
17519   364.916656 0.000000
17520   364.958313 0.000000
17521   364.958344 0.000000
17522   365.000000 0.000000
17523
17524   PP  2  17520
17525   0.000000 0.000000
17526   0.041660 0.000000
17527   0.041670 0.000000
17528   0.083320 0.000000
17529   0.083330 0.000000
17530   0.124990 0.000000
17531   0.125000 0.000000
17532   0.166660 0.000000
17533   0.166670 0.000000
17534   0.208320 0.000000
17535   0.208330 0.000000
17536   0.249990 0.000000
17537   0.250000 0.000000
17538   0.291660 0.000000
17539   0.291670 0.000000
17540   0.333320 0.000000
17541   0.333330 0.000000
17542   0.374990 0.000000
17543   0.375000 0.006374
17544   0.416660 0.006374
17545   0.416670 0.014602
17546   0.458320 0.014602
17547   0.458330 0.022003
17548   0.499990 0.022003
17549   0.500000 0.039312
17550   0.541660 0.039312
```

870
871 *Figure 9.15 Screenshot of '.forc': precipitation data*

872

```
630790   364.958313 205.330002
630791   364.958344 203.679993
630792   365.000000 203.679993
630793
630794   LAI  1  24  0.0002
630795   0       0.527609
630796   29      0.527609
630797   30      0.552859
630798   59      0.552859
630799   60      0.600613
630800   89      0.600613
630801   90      0.64426
630802   119     0.64426
630803   120     0.724488
630804   149     0.724488
630805   150     0.836395
630806   179     0.836395
630807   180     0.854004
630808   209     0.854004
630809   210     0.812654
630810   239     0.812654
630811   240     0.72533
630812   269     0.72533
630813   270     0.632919
630814   299     0.632919
630815   300     0.562581
630816   329     0.562581
630817   330     0.530051
630818   365     0.530051
630819
630820   LAI  2  24  0.0002
630821   0   8.7600
630822   30   8.7600
630823   31   9.1600
630824   58   9.1600
630825   59   9.8270
630826   89   9.8270
630827   90  10.0930
630828   119  10.0930
630829   120  10.3600
630830   150  10.3600
630831   151  10.7600
630832   180  10.7600
630833   181  10.4930
630834   211  10.4930
630835   212  10.2270
630836   242  10.2270
630837   243  10.0930
630838   272  10.0930
630839   273   9.8270
630840   303   9.8270
630841   304   9.1600
```

Figure 9.16 Screenshot of '.forc': LAI data

```
631517   304    0.2096
631518   333    0.2096
631519   334    0.1948
631520   365    0.1948
631521
631522   MF   1   2
631523   0    0.0
631524   365    0.0
631525
```

Figure 9.17 Screenshot of '.forc': MF data

9.13 projectName File

You may just write the name of the project in the projectName.txt. Say you write "sc", then the PIHM will look for all inputfiles beginning with "sc.".

10 Running PIHM

10.1 PIHM codes

The most recent version of the PIHM code can be obtained from Dr. Mukesh Kumar (mukesh.kumar@ua.edu). Older version are available from the PIHM website (http://www.pihm.psu.edu/pihm_downloads.html) under **Downloads** tab and the SourceForge website for PIHM (https://sourceforge.net/projects/pihmmodel/). These websites include PIHM version v2.2 and v2.0, respectively. PIHM is written in C programming language, so you need gcc to compile this program. Sundials is used for solver in PIHM, so you need to install Sundials before compiling PIHM. For the current version of PIHM, Sundials v2.4.0 is required. Compiling PIHM is managed by 'makefile' that you can compile with 'make pihm' and you can erase configurations and executive file with 'make clean'. With 'makefile', you can select options such as coupled, decoupled, neglecting subsurface flow, and so on.

10.2 Sundials

Sundials (Suite of Nonlinear and Differential/Algebraic Equation Solvers; https://computation.llnl.gov/projects/sundials) is implemented with the goal of providing robust time integrators and nonlinear solvers that can easily be incorporated into existing simulations codes. The primary design goals are to require minimal information from the user, allow users to easily supply their own data structures underneath the packages, and allow for easy incorporation of user-supplied linear solvers and preconditioners. Sundials consists of six different solvers. For more details, you can see in the official website with the link attached above.

For PIHM, numerical calculations are conducted by functions in the package of Sundials. Therefore, Sundials should be installed before compiling PIHM. After downloading the compressed file for Sundials v2.4.0, you can follow commands shown below on a *NIX system. You should know two things. First, you can't set the prefix path as same as the uncompressed path.

```
% tar xzf sundials-2.4.0.tar.gz
% cd sundials-2.4.0.tar.gz
% ./configure –prefix=/path_to_dir/
% make
% make install
```

10.3 Compiling PIHM using Makefile

To compile PIHM, you should check two things in the 'makefile': path of prefix (of Sundials) and running option. The path of prefix you used when installing Sundials is used in this 'makefile' to load Sundials package. The running options include (w/sub & surf, wo/surf, wo/unsat), number of layers (2 or 3), method of wave approximation (kinematic or diffusive wave), and so on.

Figure 10.1 'makefile' for PIHM

After adjusting 'makefile' to your path of Sundials and running option, you can compile PIHM with following commands.

```
% make clean
% make pihm
```

Then, you would get executive file, 'pihm', in current folder. It is shown in Fig. 10.2.

Figure 10.2 Compiled executive file, 'pihm'

10.4 Running PIHM

To run PIHM, you need a simple command written below.

```
% ./pihm
```

PIHM will read characters in 'projectName.txt', and will read files named by such project name, such as '{projectname}.mesh', '{projectname}.att', and so on.

The project name of input files generated during this tutorial is 'test02', and running PIHM of project name 'test02' is shown below in Fig. 10.3.

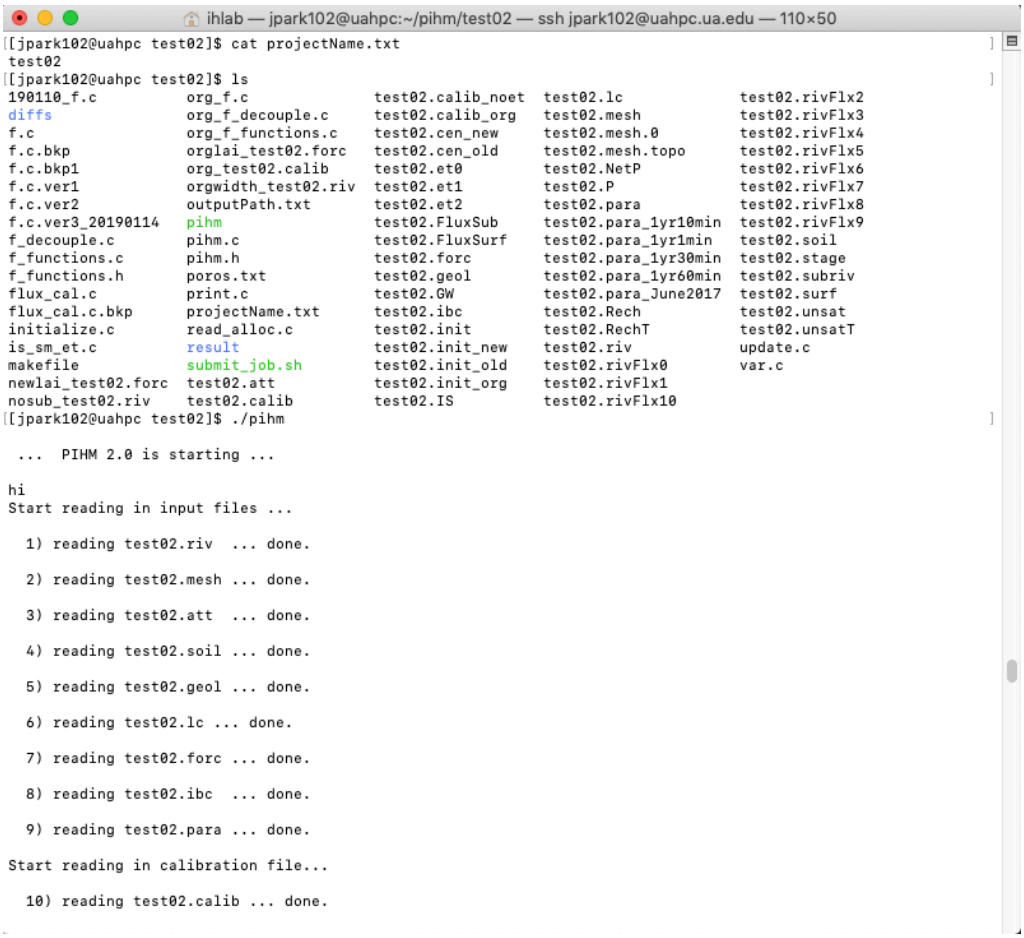


Figure 10.3 Running PIHM with project name 'test02'

10.5 Additional tips

If you want to see what parameter values are applied, then you can modify the last part of 'initialize.c' to print initialized values on the terminal. It is hard and time consuming to obtain parameters and properties of each element using GIS. Rather than that, you can modify program code to print what values were inserted in the model run.

The initializing code, 'initialize.c' in the current version of PIHM, includes commented sentences to print area, porosity, and many parameters and properties of each element and river segment. You can simply uncomment them and use to print these values. If you modify codes, then you should clean the executive file ('make clean') and compile again ('make pihm') before running PIHM.

References

Bhatt, G., Kumar, M., & Duffy, C. J. (2014). A tightly coupled GIS and distributed hydrologic modeling framework. *Environmental modelling & software*, *62*, 70-84.

Kumar, M., Bhatt, G., & Duffy, C. J. (2009). An efficient domain decomposition framework for accurate representation of geodata in distributed hydrologic models. *International Journal of Geographical Information Science*, *23*(12), 1569-1596.

Kumar, M., Bhatt, G., & Duffy, C. J. (2010). An object-oriented shared data model for GIS and distributed hydrologic models. *International Journal of Geographical Information Science*, *24*(7), 1061-1079.

GLCF, n.d.: UMD Land Cover Classification. Retrieved from http://glcf.umd.edu/data/landcover/

Hydrology Group of PSU, n.d.: PIHMgis website. Retrieved from http://www.pihm.psu.edu/pihmgis_home.html

Hydrology Group of PSU, n.d.: PIHM wetsite. Retrived from http://www.pihm.psu.edu/pihm_downloads.html

Hydrology Group of PSU, 2007: Input file format manual of PIHM v2.0. Retrieved from http://www.pihm.psu.edu/Downloads/Doc/pihm2.0_input_file_format.pdf

Hydrology Group of PSU, 2010: Tutorial of V-Catchment Watershed for PIHMgis v2.3. Retrieved from http://www.pihm.psu.edu/PIHMgis_v2.3_Tutorial_VCatchment.pdf

Hydrology Group of PSU, 2015: SourceForge website for PIHM. Retrieved from https://sourceforge.net/projects/pihmmodel/

Kumar, M. (2009). Toward a hydrologic modeling system, https://etda.libraries.psu.edu/files/final_submissions/4388

Lawrence Livermore National Laboratory, n.d.: Sundials website. Retrieved from https://computation.llnl.gov/projects/sundials

MRLC, n.d.: National Land Cover Database 2011 (NLCD2011) Legend. Retrieved from https://www.mrlc.gov/data/legends/national-land-cover-database-2011-nlcd2011-legend

NASA GES DISC, 2018: READEME document for North American Land Data Assimilation System Phase 2 (NLDAS-2) Product. Retrieved from https://hydro1.gesdisc.eosdis.nasa.gov/data/NLDAS/README.NLDAS2.pdf

NASA GES DISC, n.d.: Data Access. Retrieved from https://disc.gsfc.nasa.gov/data-access

NASA GES DISC, n.d.: NLDAS_FORA0125_H: NLDAS Primary Forcing Data L4 Hourly 0.125 x 0.125 degree V002. Retrieved from https://disc.gsfc.nasa.gov/datasets/NLDAS_FORA0125_H_V002/summary

NASA LDAS, n.d.: NLDAS Unified Mask. Retrieved from https://ldas.gsfc.nasa.gov/nldas/asc/NLDASmask_UMDunified.asc

NASA LDAS, n.d.: NLDAS Vegetation Parameters. Retrieved from https://ldas.gsfc.nasa.gov/nldas/NLDASmapveg.php

Qu, Y., 2005: An integrated hydrologic model for multi-process simulation using semi-discrete finite volume approach. Ph.D diss., Pennsylvania State University.

Qu, Y., Duffy, C.J., 2007: A semidiscrete finite volume formulation for multiprocess watershed simulation. Water Resources Research, 43, W08419, doi:10.1029/2006WR005752.

UC Davis California Soil Resource Lab, n.d.: SoilWeb_nc097. Retrieved from https://casoilresource.lawr.ucdavis.edu/soil_web/ssurgo.php?action=list_mapunits&areasymbol=nc097

UC Davis California Soil Resource Lab, n.d.: SoilWeb_nc159. Retrieved from https://casoilresource.lawr.ucdavis.edu/soil_web/ssurgo.php?action=list_mapunits&areasymbol=nc159

USDA NRCS, n.d.: Soil Data Development Toolbox. Retrieved from https://www.nrcs.usda.gov/wps/portal/nrcs/detail/soils/survey/geo/?cid=nrcs142p2_053628

USDA NRCS, n.d.: Soil Data Viewer. Retrieved from https://www.nrcs.usda.gov/wps/portal/nrcs/detailfull/soils/survey/geo/?cid=nrcs142p2_053620

USDA NRCS, n.d.: Web Soil Survey. Retrieved from https://websoilsurvey.sc.egov.usda.gov/App/WebSoilSurvey.aspx

USGS, n.d.: TNM Download (V1.0). Retrieved from https://viewer.nationalmap.gov/basic/

Wang, D., Liu, Y., & Kumar, M. (2018). Using nested discretization for a detailed yet computationally efficient simulation of local hydrology in a distributed hydrologic model. *Scientific reports*, *8*(1), 5785.

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

Appendix A: Generating forcing file from NLDAS-2 grib

Additionally, in this version of tutorial manual, two program codes are introduced to extract data from NLDAS-2 grib files and organize data into the format of '.forc' file, respectively.

First is a C++ code, 'extractForcing.cpp'. This code opens NLDAS-2 grib files and saves information of target grids and target duration into text files. This code consists 'main', 'stepOneHour', and 'MainFunction'. To set target grids and target duration, you should modify two things. First, you should modify lines shown in Fig. A.1. There should be the number of target grids, x-y grid index, and target duration. Second, you should find sentences defining characters of 'NLDAS_FOLDER' and 'OUT_FOLDER', and modify them for your folder paths. Compiling this code ('gcc extractForcing.cpp -o a.out') and running it will result in text files shown in Fig. A.2.

Figure A.1 Target grids and duration in 'extractForcing.c'

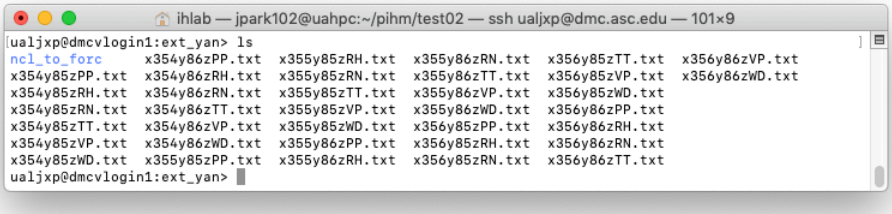


Figure A.2. Result files using 'extractForcing.c'

Second code is a ncl code, 'make_forc_1.ncl'. It reads extracted text files and additional files for monthly LAI and roughness length

values, and generates ‘sample.txt’ that includes all forcing and vegetation time-series information within ‘.forc’ format. To generate the result file, you should modify two things. First, as shown in Fig. A.3, there are grid numbers, length of duration, and paths. Grid number should be same with previous code, and length of duration should be hours. Paths should be modified in whole code.

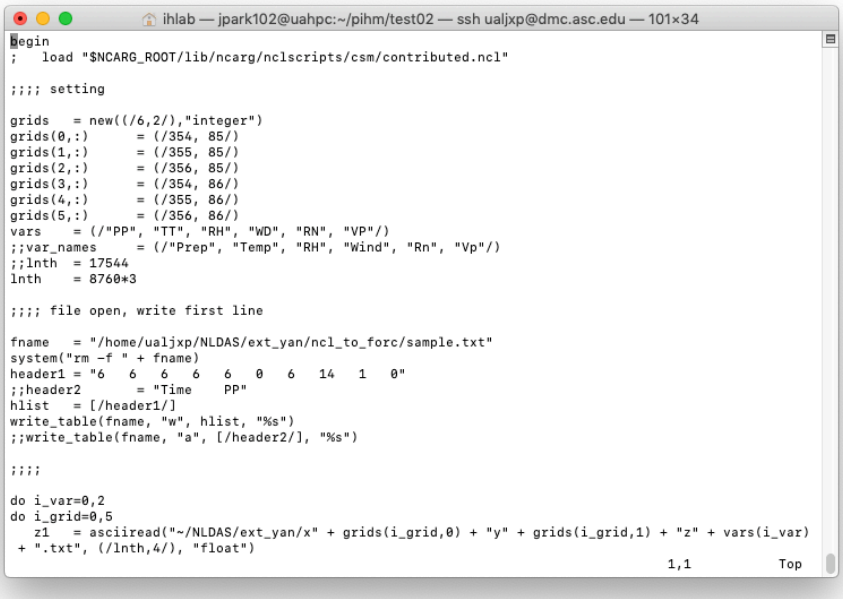


Figure A.3 Grid and duration setting in 'make_forc_1.ncl'

Second, as shown in Fig. A.4, you should modify settings for vegetation parameters, LAI and RL. Additional files, ‘rl_nldas.txt’ and ‘lai_nldas.txt’ includes LAI and RL values for 12-month columns and 14-land-cover rows. Check the paths of these vegetation parameter files are right. You can also modify ‘tstep_veg’, the number of records, and ‘jday’, time for each record. Setting in Fig. A.4 is for 3-year forcing file. You can change in any values, but you should check first ‘jday’ value is 0 and the last ‘jday’ value is the last day’s Julian date.

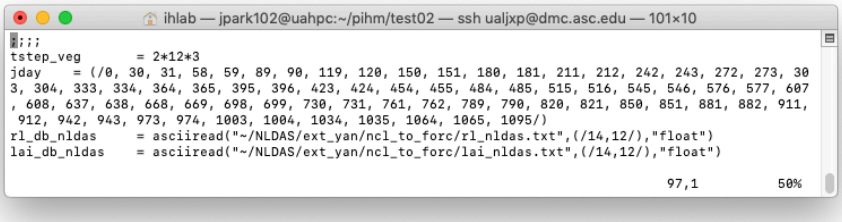


Figure A.4 Setting for vegetation parameters in 'make_forc_1.ncl'

Appendix B: Water balance

The program code 'waterbudget_alldomain.c' is for calculating area-averaged water balance using model output files and some additional information (area and porosity of elements, and area of river segments). To use this code, you should check three things. First, for reading model output files, the executive file of this code should be run in the same folder with output files. Of course, change the project name in file names. In Fig. B.1, you can see that output files are opened without any additional paths. Second, for reading additional information extracted in section 10.5, these files' path should be modified in the code. Finally, reference values (NumEle, NumTS, Interval, NumRiv, and RivOutlet) should be modified. 'RivOutlet' is the number of outlet river segment that have '-3' value (it means discharge flows outside) in 4th column of '.riv' file. After compiling and running this code, you would get daily, spatially-averaged water balance table ('{projectname}.wb').

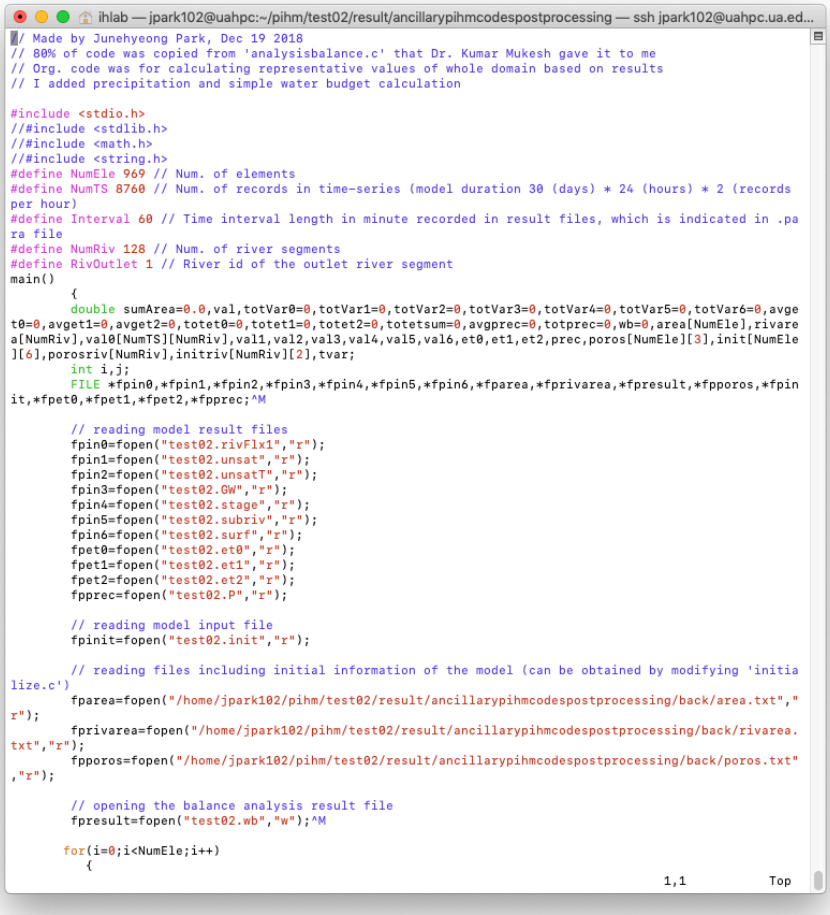
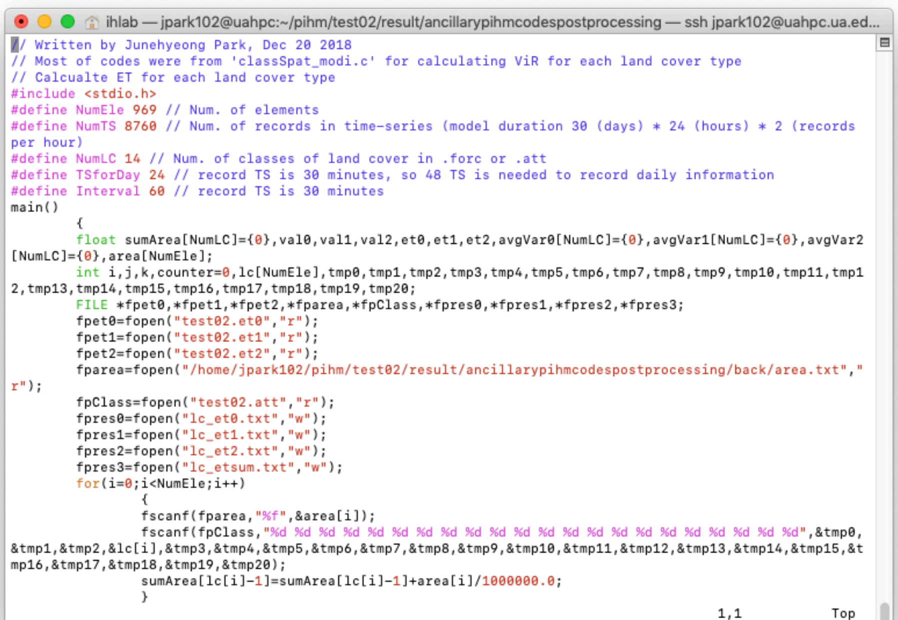


Figure B.1 Water balance program code 'waterbudget_alldomain_1.c'

Appendix C: Evapotranspiration for each land cover type

The program code 'lc_et.c' shown in Fig. C.1 is for calculating area-averaged evapotranspiration for each land cover type. This information can be useful to check how the ET calculation is going on in model, and land cover values are applied correctly, simultaneously. Similar to water balance program code in Appendix B, this program code also need to be aware of three things. First, check the output file names (especially project name), and put executive file in same folder with these output files. Second, check the path of 'area.txt', which includes area of each element. Finally, reference values (NumEle, NumTS, NumLC, TSforDay, and Interval) should be modified for each model run. Compiling and running this program code, you would get daily, area-averaged evapotranspiration for ET0 (interception loss), ET1 (transpiration), ET2 (evaporation from ground), and their summation.



```
ihlab — jpark102@uahpc:~/pihm/test02/result/ancillarypihmcodespostprocessing — ssh jpark102@uahpc.ua.ed...
// Written by Junehyeong Park, Dec 20 2018
// Most of codes were from 'classSpat_modi.c' for calculating ViR for each land cover type
// Calcualte ET for each land cover type
#include <stdio.h>
#define NumEle 969 // Num. of elements
#define NumTS 8760 // Num. of records in time-series (model duration 30 (days) * 24 (hours) * 2 (records
per hour)
#define NumLC 14 // Num. of classes of land cover in .forc or .att
#define TSforDay 24 // record TS is 30 minutes, so 48 TS is needed to record daily information
#define Interval 60 // record TS is 30 minutes
main()
        {
        float sumArea[NumLC]={0},val0,val1,val2,et0,et1,et2,avgVar0[NumLC]={0},avgVar1[NumLC]={0},avgVar2
[NumLC]={0},area[NumEle];
        int i,j,k,counter=0,lc[NumEle],tmp0,tmp1,tmp2,tmp3,tmp4,tmp5,tmp6,tmp7,tmp8,tmp9,tmp10,tmp11,tmp1
2,tmp13,tmp14,tmp15,tmp16,tmp17,tmp18,tmp19,tmp20;
        FILE *fpet0,*fpet1,*fpet2,*fparea,*fpClass,*fpres0,*fpres1,*fpres2,*fpres3;
        fpet0=fopen("test02.et0","r");
        fpet1=fopen("test02.et1","r");
        fpet2=fopen("test02.et2","r");
        fparea=fopen("/home/jpark102/pihm/test02/result/ancillarypihmcodespostprocessing/back/area.txt","
r");
        fpClass=fopen("test02.att","r");
        fpres0=fopen("lc_et0.txt","w");
        fpres1=fopen("lc_et1.txt","w");
        fpres2=fopen("lc_et2.txt","w");
        fpres3=fopen("lc_etsum.txt","w");
        for(i=0;i<NumEle;i++)
                {
                fscanf(fparea,"%f",&area[i]);
                fscanf(fpClass,"%d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d",&tmp0,
&tmp1,&tmp2,&lc[i],&tmp3,&tmp4,&tmp5,&tmp6,&tmp7,&tmp8,&tmp9,&tmp10,&tmp11,&tmp12,&tmp13,&tmp14,&tmp15,&t
mp16,&tmp17,&tmp18,&tmp19,&tmp20);
                sumArea[lc[i]-1]=sumArea[lc[i]-1]+area[i]/1000000.0;
                }
                                                                                    1,1           Top
```

Figure C.1 Program code 'lc_et.c' to calculate avg. evap. for each land cover type