

PennState Integrated Hydrologic model (PIHM) Tutorial¹

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 Contents²

1. Introduction	1
2. Download PIHMgis	1
3. Download target watershed and related raw data	2
3.1 Target watershed	2
3.2 Data Requirement	2
4. PIHMgis Project	3
4.1 New (PIHMgis) Project	4
4.2 Open Project	4
4.3 Import Project	5
4.4 Close Project	6
5. Raster Processing	6
5.1 Introduction	6
5.2 Fill Pits	9
5.3 Flow Grids	10
5.4 Stream Grids	11
5.5 Link Grids	12
5.6 Catchment Grids	14
5.7 Stream Polyline	14
5.8 Catchment Polygon	15
5.9 Summary	16
6. Vector Processing	17
6.1 Dissolve Polygons	17
6.2 Polygon to Polylines	18
6.3 Simplify Lines	19
6.4 Polyline to Split Lines and modifications	20
6.5 Vector Merge	21
7. Domain Decomposition	23
7.1 Read Topology	23

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

7.2	Triangulation	23
7.3	TIN Shape Layer	25
8.	Pre-processing raw data	26
8.1	Introduction	26
8.2	Soil Data	27
8.3	Land Cover Data	35
8.4	Forcing Data	37
9.	DataModel Loader	44
9.1	Introduction	44
9.2	Mesh Data File	44
9.3	Att Data File	46
9.4	Riv Data File	49
9.5	Soil Data File	53
9.6	Geol Data File	56
9.7	LC Data File	57
9.8	Init Data File	59
9.9	IBC Data File	60
9.10	Param Data File	63
9.11	Calib Data File	66
9.12	Forc Data File	67
9.13	projectName File	74
10.	Running PIHM	74
10.1	PIHM codes	74
10.2	Sundials	74
10.3	Compiling PIHM using Makefile	75
10.4	Running PIHM	77
10.5	Additional tips	79

Appendix A: Generating forcing file from NLDAS-2 grib

Appendix B: Water balance

Appendix C: Evapotranspiration for each land cover type

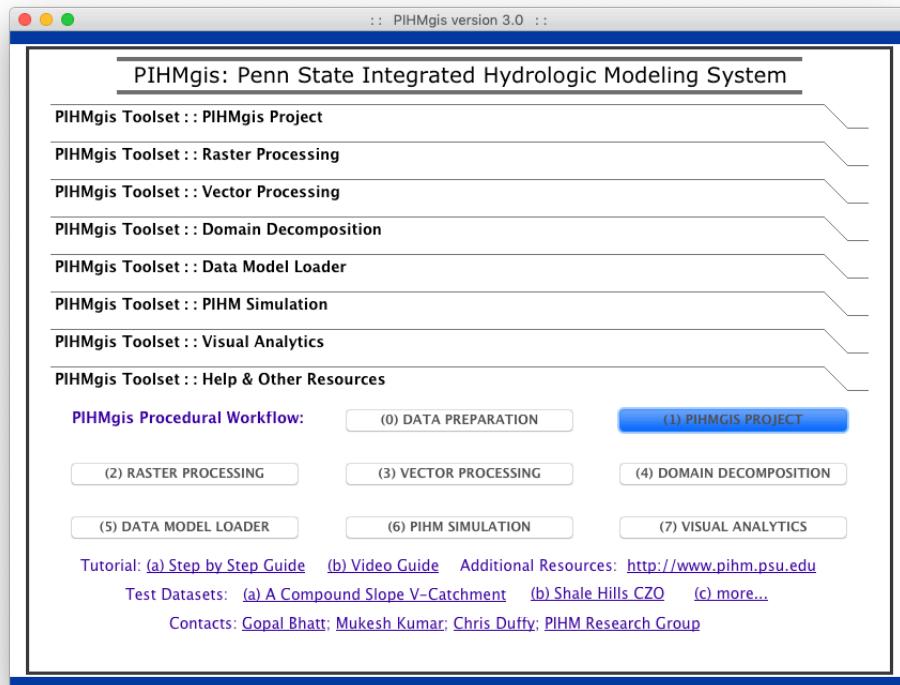
1 1 Introduction

2 PIHM (Penn State Integrated Hydrologic Model) is an open-source,
3 physically-based, fully-distributed hydrologic model (Qu, 2005; Qu and
4 Duffy, 2007; Kumar, 2009). The model uses unconstructed mesh for
5 discretizing the model domain, which provides numerous advantages in
6 terms of representational accuracy (Kumar et al., 2009) and
7 computational efficiency (Wang et al., 2018). Running the model
8 requires populating each structured mesh with meteorological and
9 physiographic properties. This is done using PIHMGis (Bhatt, et al.,
10 2014), an open-source, platform independent, extensible and tightly-
11 coupled GIS interface to PIHM. PIHMGis uses a shared data model
12 framework (Kumar et al., 2010) that allows quick generation and
13 revision of the PIHM model input files.

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

19 2 Download PIHMGis

20 PIHMGis version 3.0 can be downloaded from the PIHMGis website
21 (http://www.pihm.psu.edu/pihmgis_home.html) under **Downloads** tab
22 (Fig. 2.1). You need to select the appropriate version according to your
23 device's OS. Users are encouraged to use the Macintosh (Mac) version.
24 Windows version has not kept up with the OS changes, and may lead to
25 crashes. If you want to run PIHMGis in Linux system, you need to
26 compile the source code natively and meet the library dependencies.
27 You may download the PIHMGis source code from GitHub. This tutorial
28 is based on the Mac version.
29



30
31

Figure 2.1 PIHMgis ver. 3.0

32

3 Download target watershed and related raw data

3.1 Target watershed

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

40 3.2 Data Requirement

41 Data needed for successfully setting up the model include:

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

47

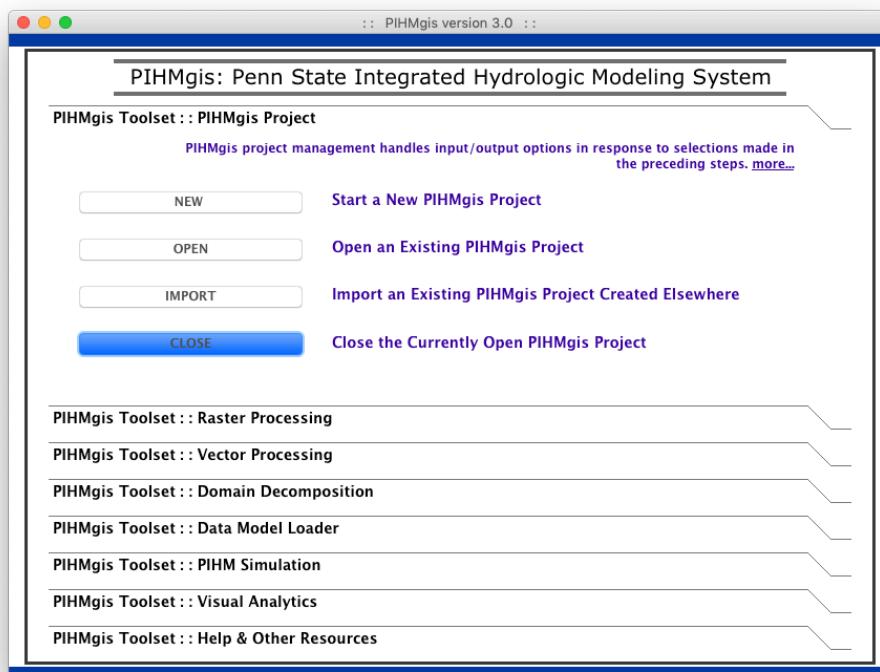
48 For this tutorial, for DEM, 'USGS NED 1 arc-second 2013 1 x 1 degree³'
49 was used. For forcing data, 'NLDAS Primary Forcing Data L4 hourly

³ TNM Download (<https://viewer.nationalmap.gov/>)

50 0.125 x 0.125 degree V002⁴ was used. For land use and vegetation,
51 'NLCD 2011 Land Cover (2011 Edition, amended 2014)⁵' and 'Monthly
52 vegetation parameters based on NLDAS land cover classes⁵' was used.
53 For soil properties, 'SSURGO (Soil Survey Geographic Database)⁶, ⁷' was
54 used. With these spatially distributed data set, you can run this model
55 with heterogeneous input layers. Footnotes provide the link to locations
56 for downloading these data for US watersheds.

57 **4 PIHMgis Project**

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



62
63

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>)

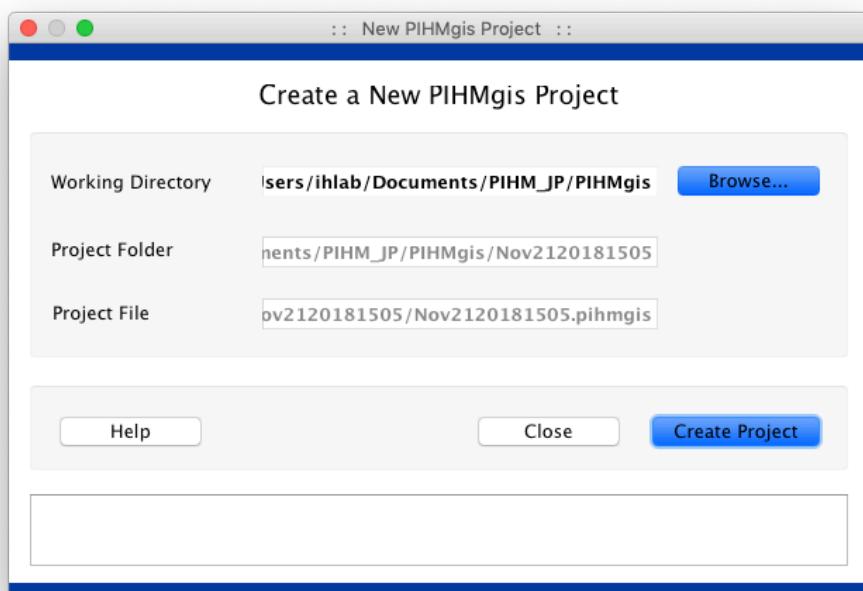
⁷ Soil Data Viewer

(https://www.nrcs.usda.gov/wps/portal/nrcs/detailfull/soils/survey/geo/?cid=nrcs142p2_053620)

64 **4.1 New (PIHMgis) Project**

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

76



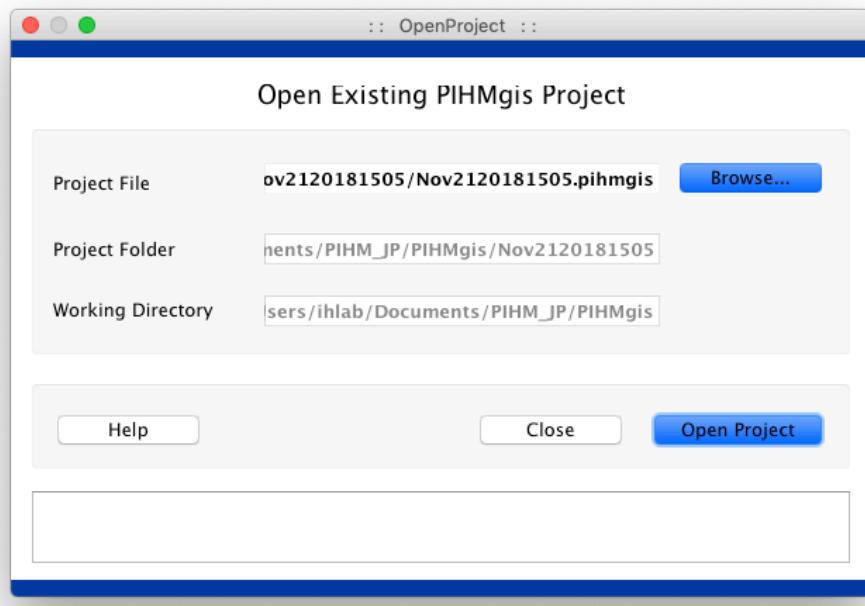
77
78
79

Figure 4.2 Create a new PIHMgis project dialog

80 **4.2 Open Project**

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

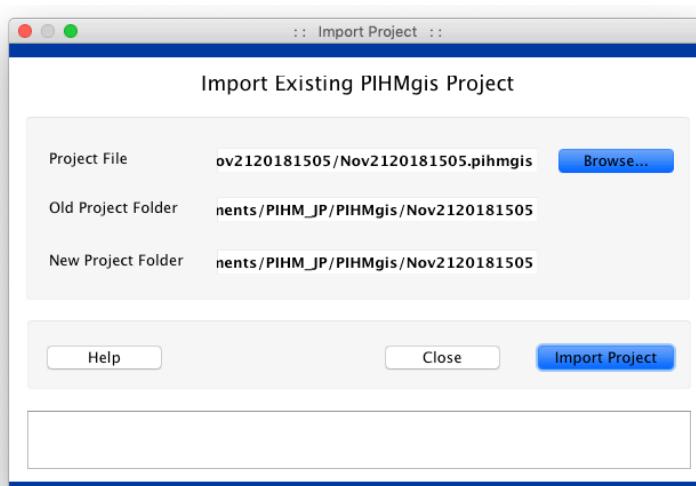
88 activities were recorded. So, you can see which modules were already
89 used, which files were inserted as input, and which files were created.
90



91
92 *Figure 4.3 Open an existing PIHMgis dialog*
93

94 **4.3 Import Project**

95 Import project module allows you to import an existing project. Since
96 the path of the file is recorded in the PIHMgis project file is an absolute
97 path – if you change the name of any directory on the root directory
98 tree those files won't be accessible. Also, if you happen to move the
99 folders around (including sending or receiving from different
100 computers), Import project module will update the existing project file
101 so that you can continue to work on the project.
102



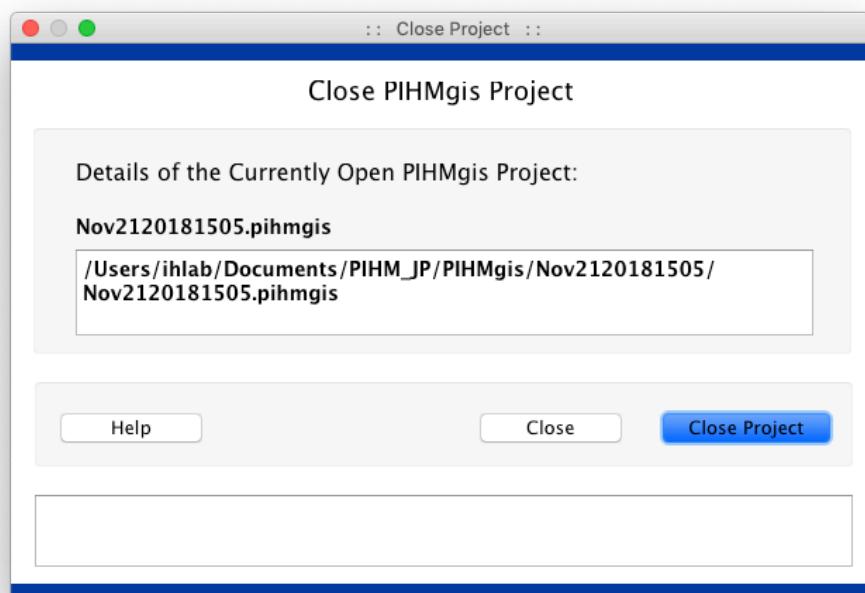
103
104 *Figure 4. Import any existing PIHMgis project*

105

106 **4.4 Close Project**

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

113



114
115
116

Figure 4.2 Close currently opened PIHMGis project

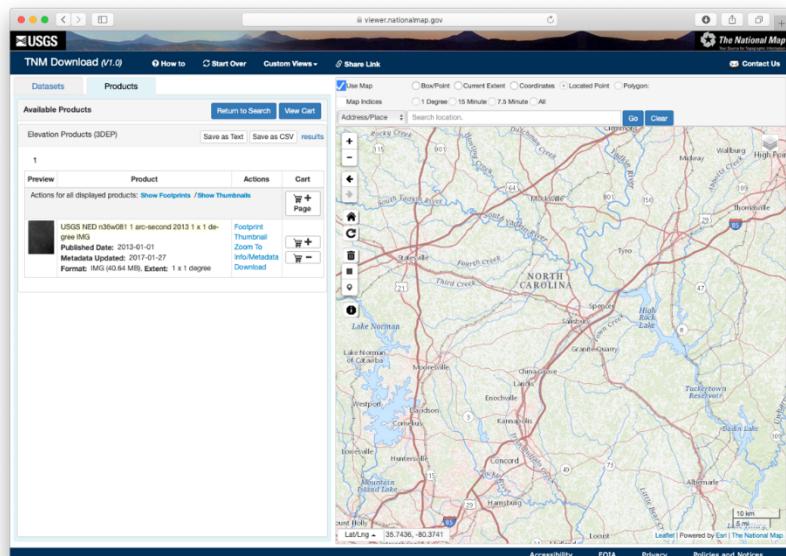
117 **5 Raster Processing**

118 **5.1 Introduction**

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

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

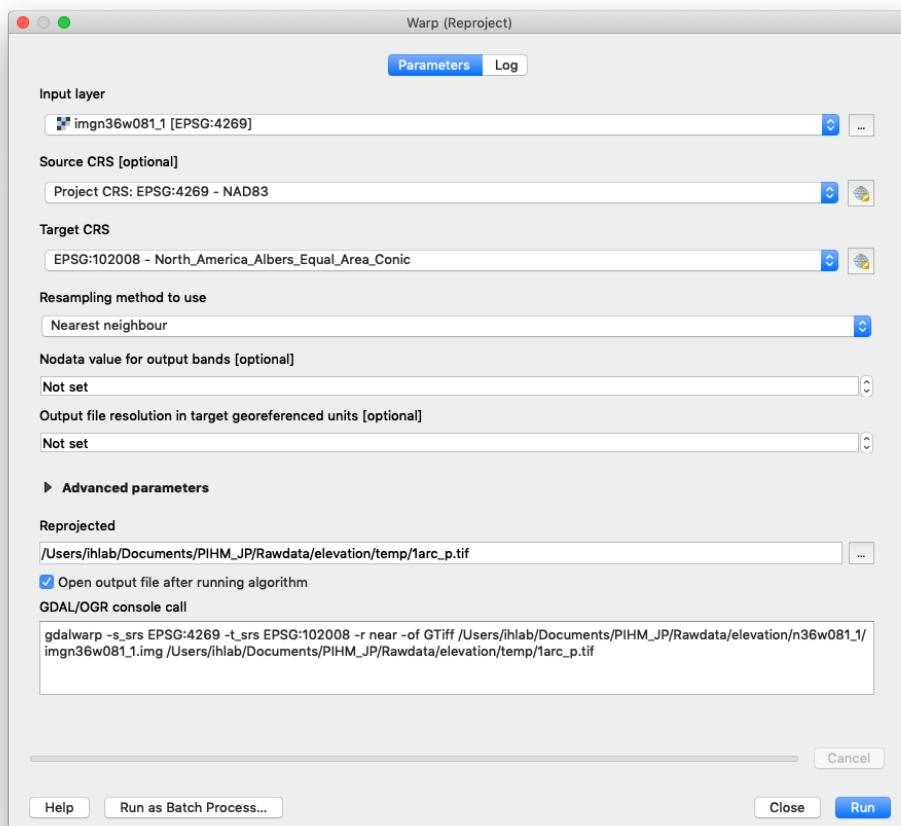
126



127
128

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

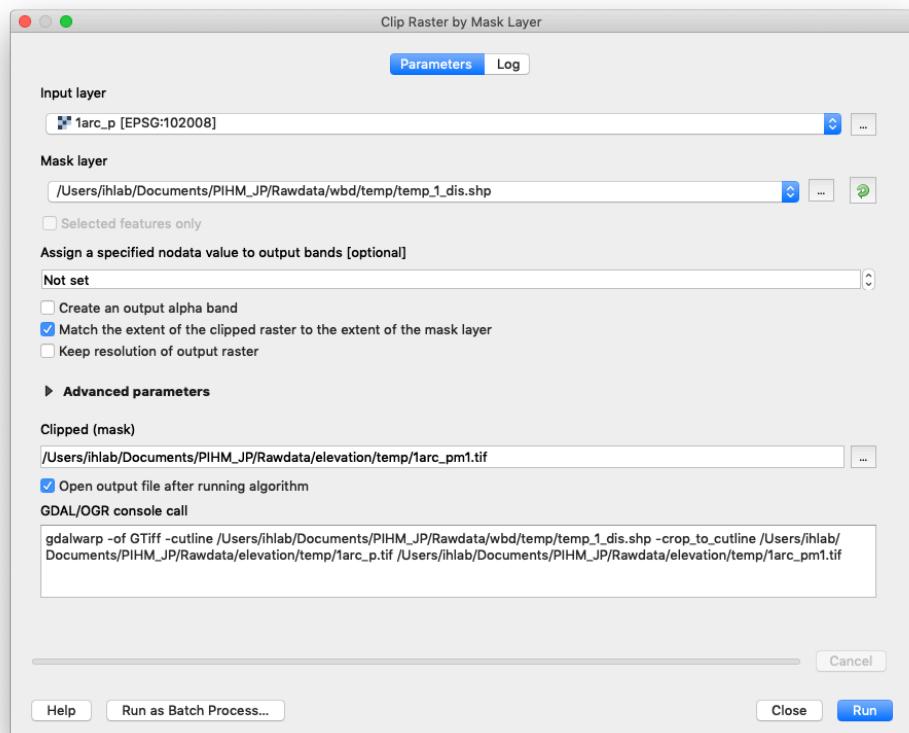
129
130



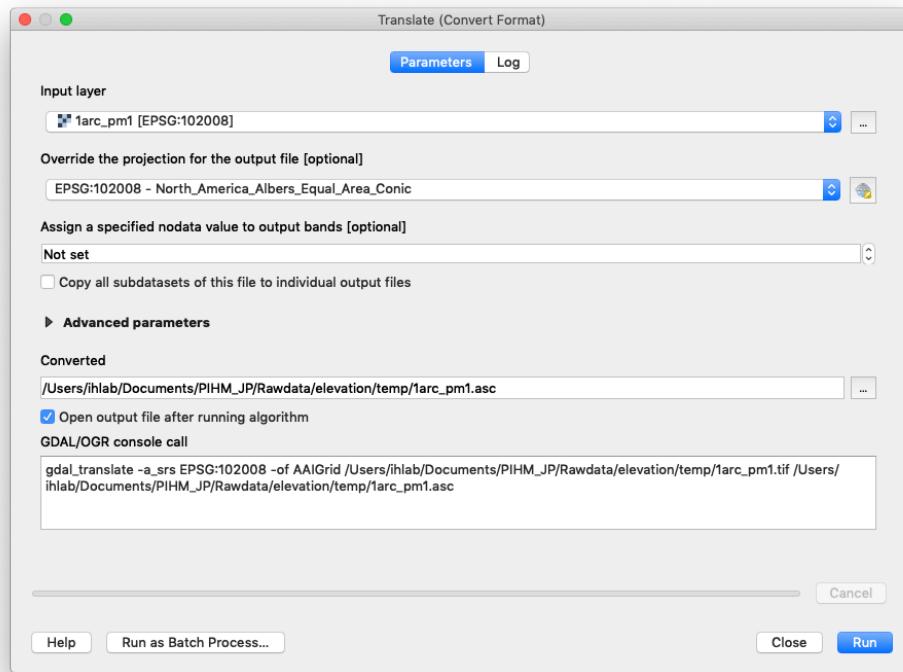
131
132

Figure 5.2 Change projection of DEM

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

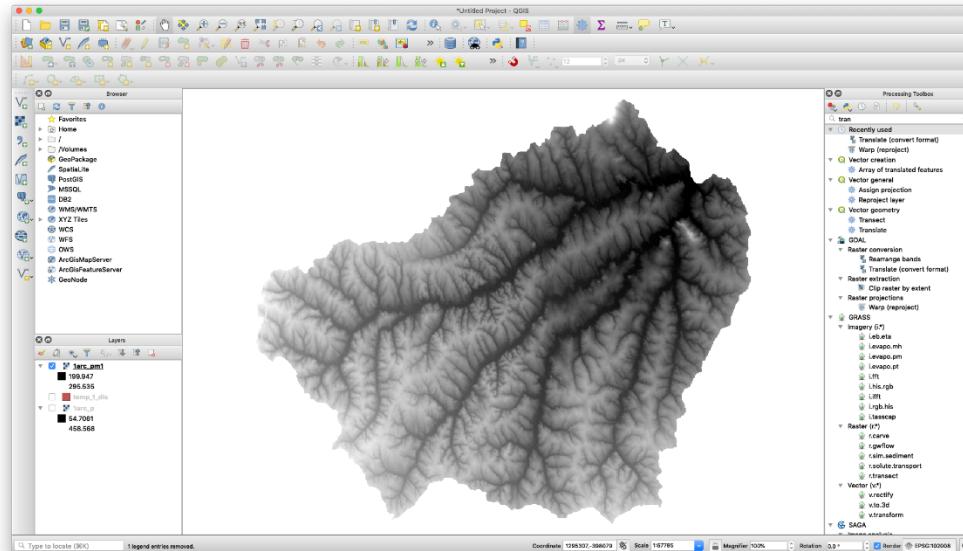


147
148 *Figure 5.3 Mask DEM with dissolved target watershed shapefile*



149
150

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



151
152
153

Figure 5.5 Results of pre-processing on DEM

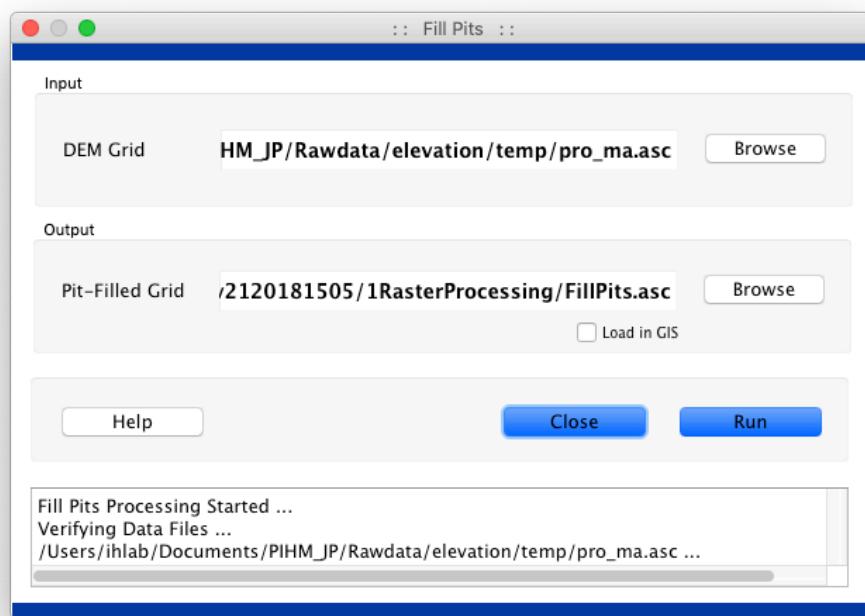
154 5.2 Fill Pits

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

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

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

171



172
173

Figure 5.6 Fill Pits dialog

174 **5.3 Flow grids**

175 Using the output file of 'Fill Pits' module, this module calculates flow
176 direction and flow accumulation. Flow direction is calculated based on
177 the D8 algorithm (O'Callaghan and Mark, 1984) that the value of each
178 grid indicates flow direction of each grid. Flow accumulation contains the
179 accumulated number of grids upstream to it for each grid using a
180 recursive procedure (Mark, 1988). Fig. 5.7 shows the flow direction and
181 flow accumulation algorithms performed on a synthetic DEM grid, and
182 Fig. 5.8 shows the dialog of 'Flow Grids' module.
183

184
185

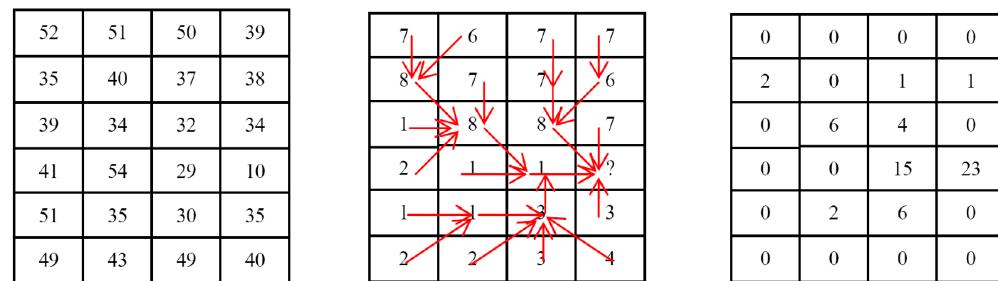
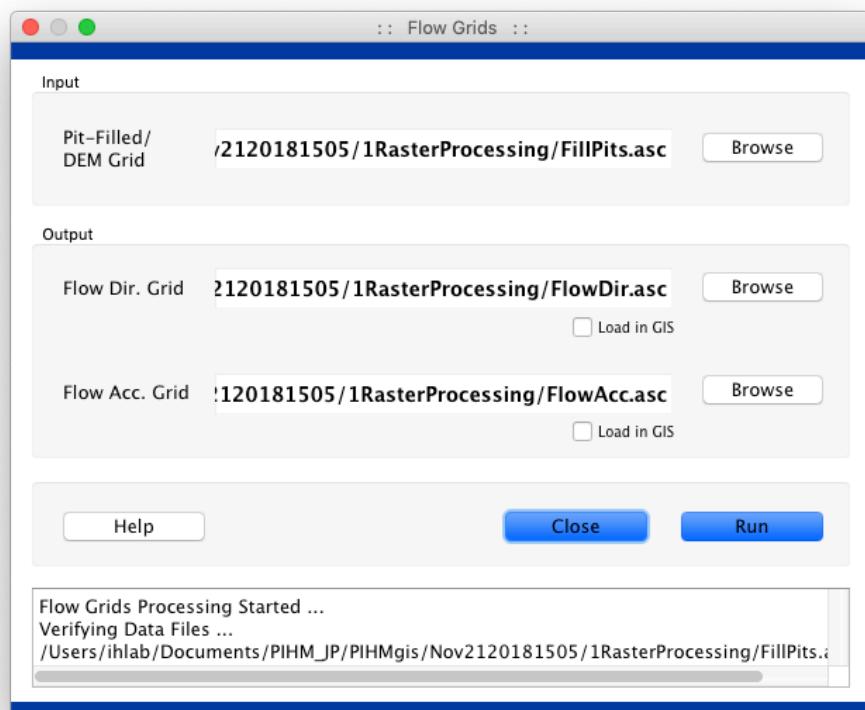


Figure 5.7 Flow direction and flow accumulation for a synthetic grid

186



187
188

Figure 5.8 Flow Grids dialog

189 5.4 Stream Grids

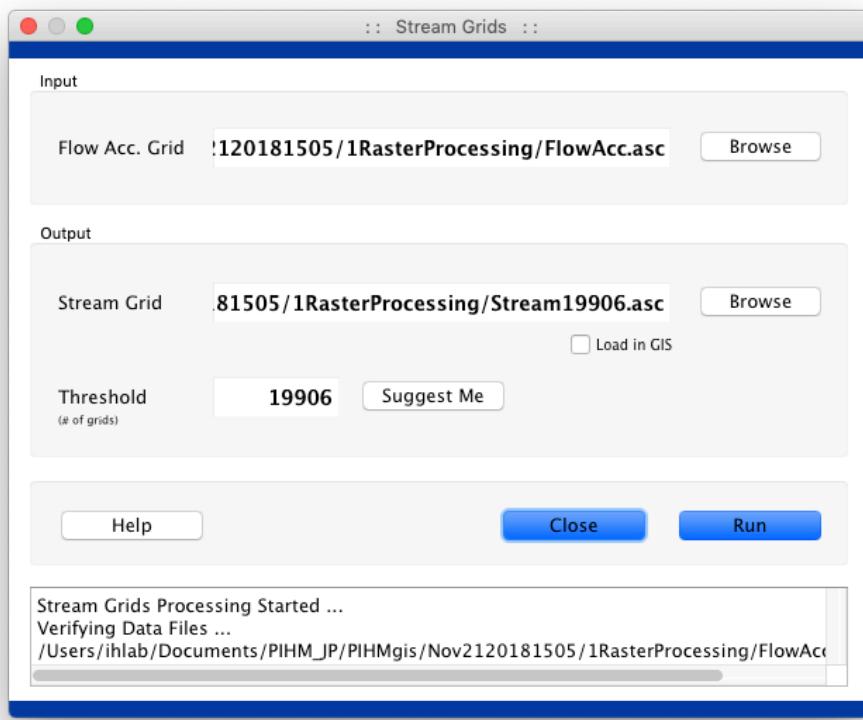
190 'Stream Grids' module marks 1 for grids which has higher value of flow
191 accumulation than the threshold (Fig. 5.9) . You can get threshold by
192 press the 'Suggest Me' button, and you can write an integer value in the
193 blank space yourself. The name of output files will contain the threshold
194 value, so you can distinguish your trials easily.
195

196
197

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

198



199
200

Figure 5.10 Stream Grids dialog

201 **5.5 Link Grids**

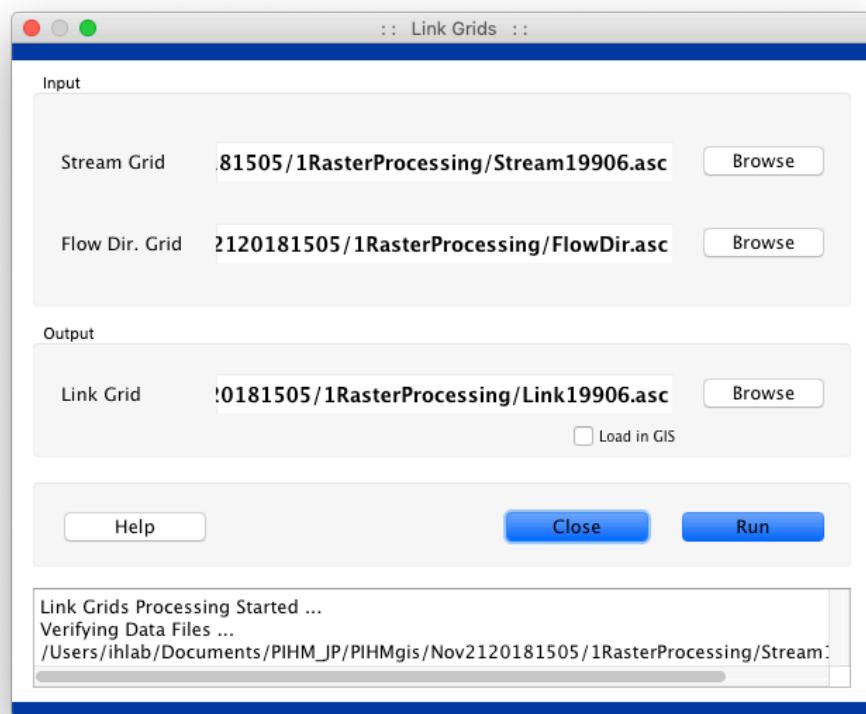
202 'Link Grids' module separates the stream grid segments at the junctions.
203 Each link grid segment is assigned a unique integer value starting with
204 1, and rest of grid assumed No Data Value. Fig. 5.11 shows the link grid
205 generated corresponding to the stream grid obtained in the previous
206 section.
207

208
209

210

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

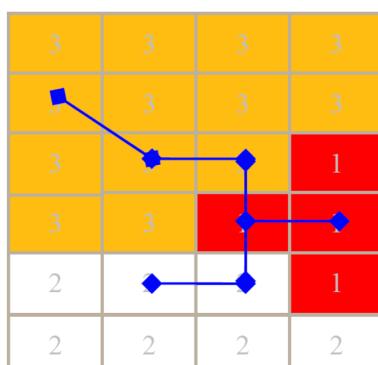
Figure 5.11 Link Grid for the synthetic grid



211
212

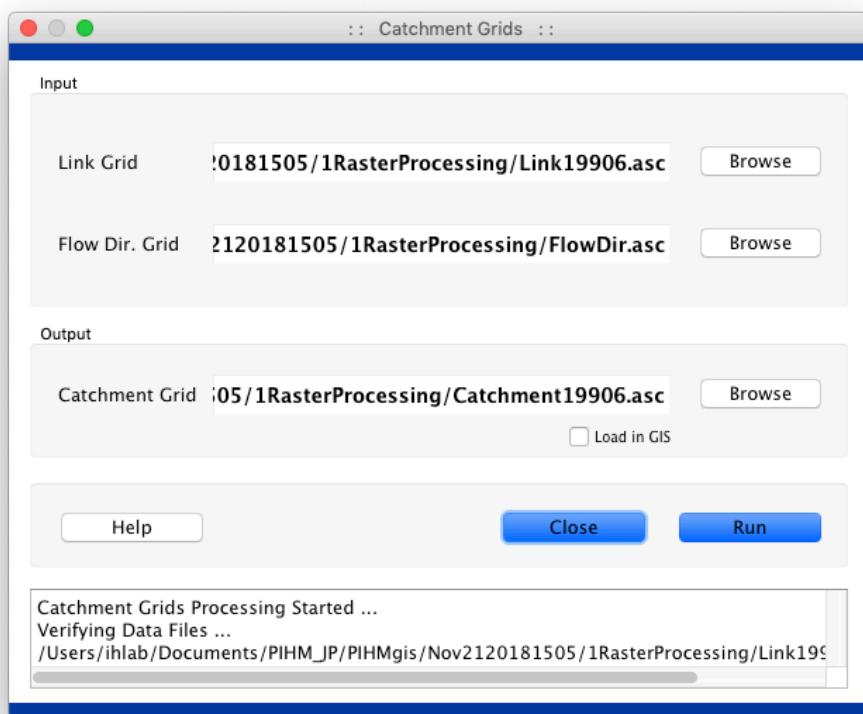
Figure 5.12 Link Grids dialog

213 **5.6 Catchment Grids**
214 All the grids draining to a particular stream segment are grouped into
215 one catchment grid. Catchment grids are marked with integer numbers
216 as same as link grid (Fig. 5.13).
217



218 *Figure 5.13 Catchment grid for the synthetic grid*
219

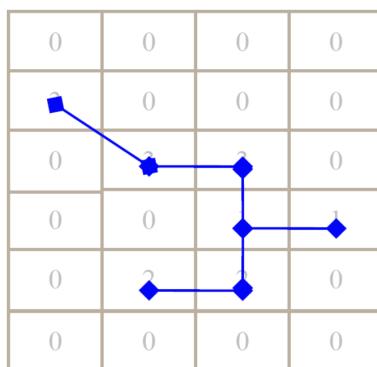
220



221 *Figure 5.14 Catchment Grids dialog*
222

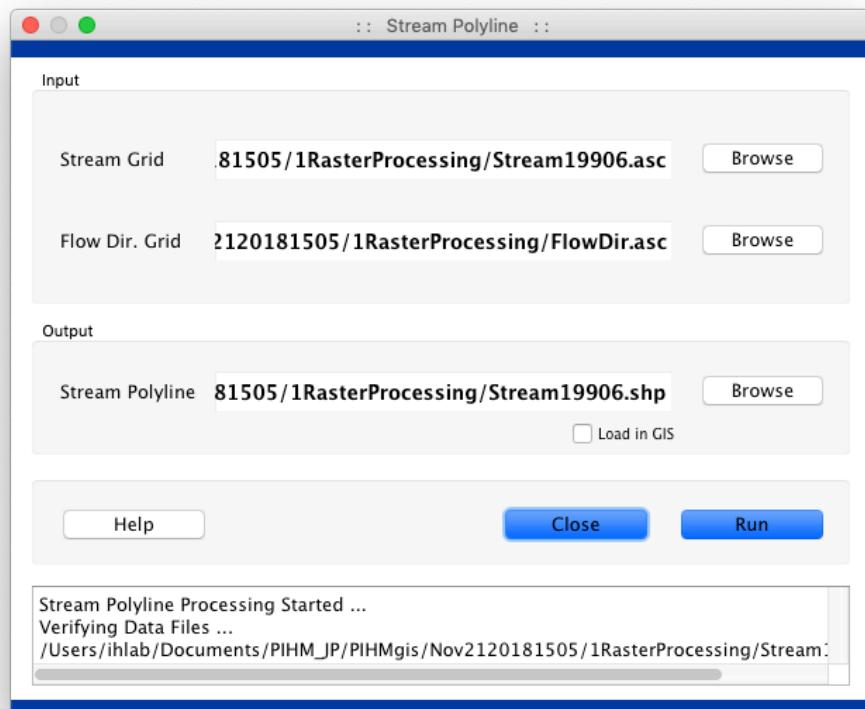
223
224 **5.7 Stream Polyline**
225 Stream polyline is the drainage network for the region of interest
226 obtained by the conversion of the link grid raster to a vector data (Fig.).

227 5.15). Each link segment forms an individual stream segment and
228 connected at the junction points. Flow direction is used to ensure that
229 the segments are topologically correct.
230



231
232 *Figure 5.15 Stream Polyline for the synthetic grid*

233



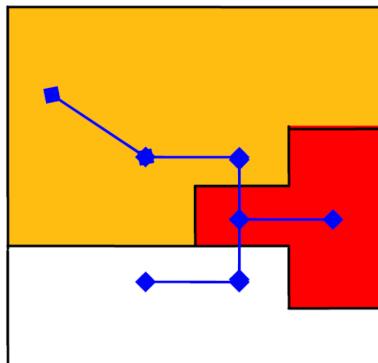
234
235 *Figure 5.16 Stream Polyline dialog*

236

237 **5.8 Catchment Polygon**

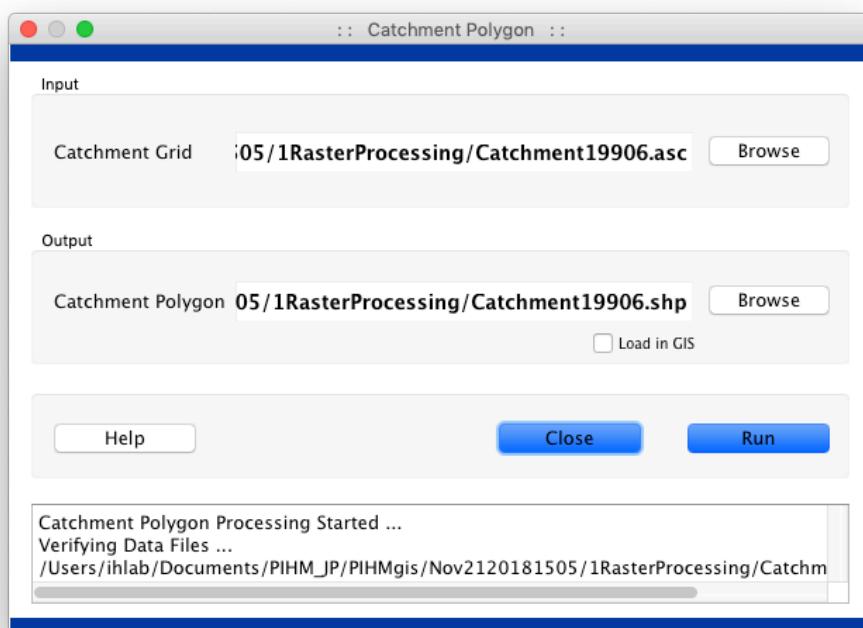
238 Catchment polygons is the vector representation of the catchment grid.
239 Similar to the catchment grid, a catchment polygon bounds the region
240 which has a single drainage outlet. Boundary of all catchment grids

241 draining to the same stream segment forms a unique catchment
242 polygon.
243



244
245 *Figure 5.17 Catchment Polygon for the synthetic grid*

246

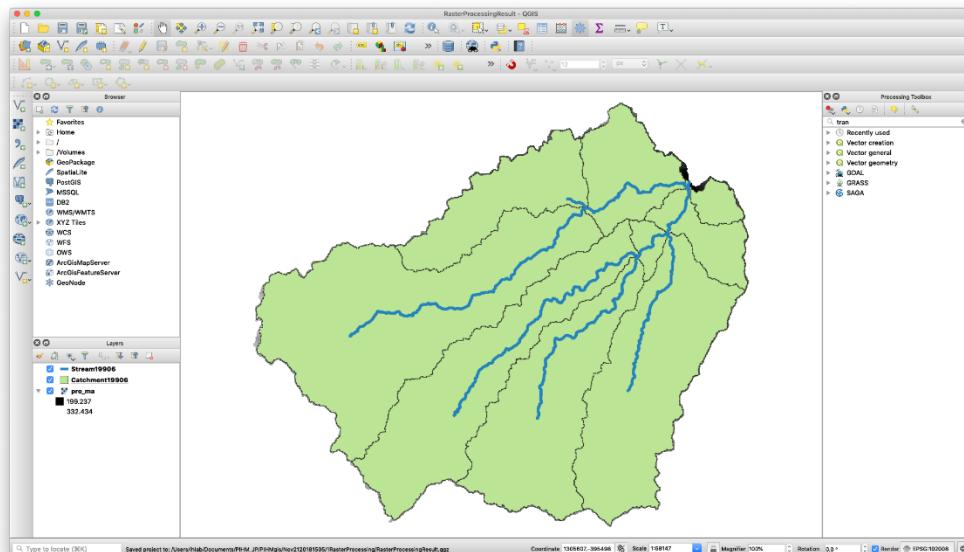


247
248 *Figure 5.18 Catchment Polygon dialog*
249

250 5.9 Summary

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

259



260
261

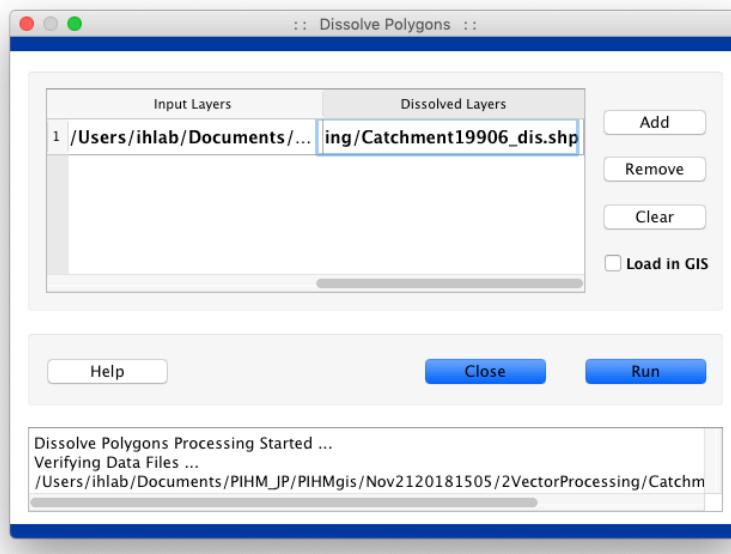
Figure 5.19 Stream polyline and catchment polygon represented in QGIS

262 6 Vector Processing

263 6.1 Dissolve Polygons

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

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



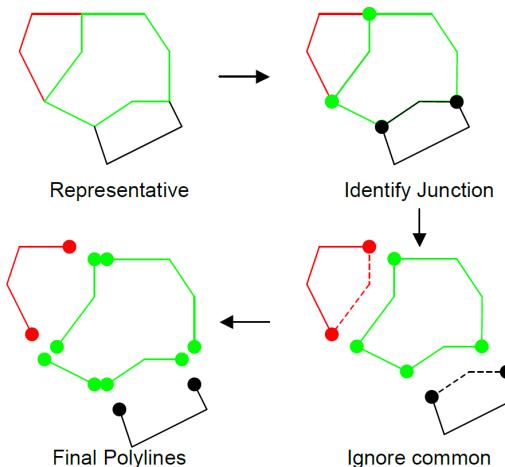
270
271

Figure 6.1 Dissolve Polygons dialog

272 **6.2 Polygon to Polylines**

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

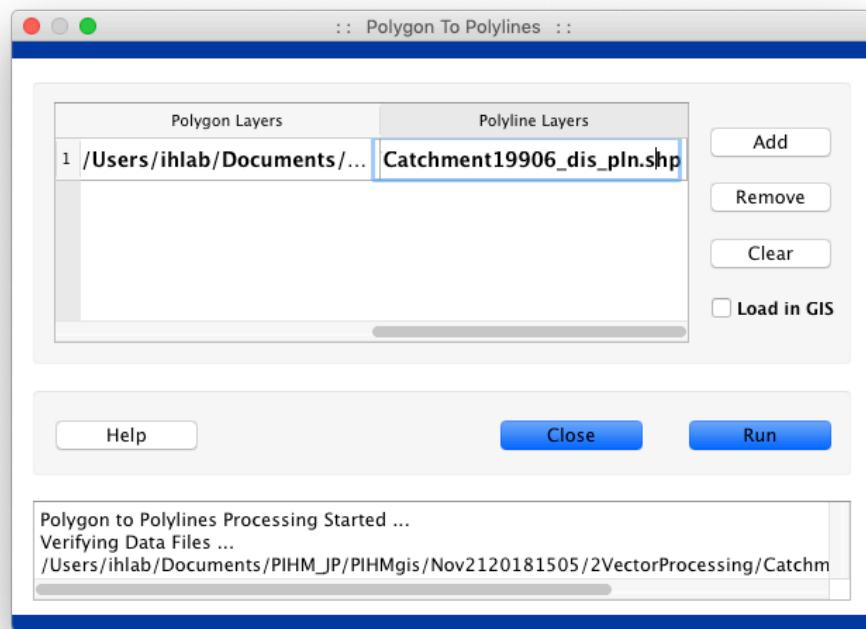
276



277

Figure 6.2 Algorithm for Polygon to Polyline

278



280

Figure 6.3 Polygon to Polylines dialog

281

282 **6.3 Simplify Lines**

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

289

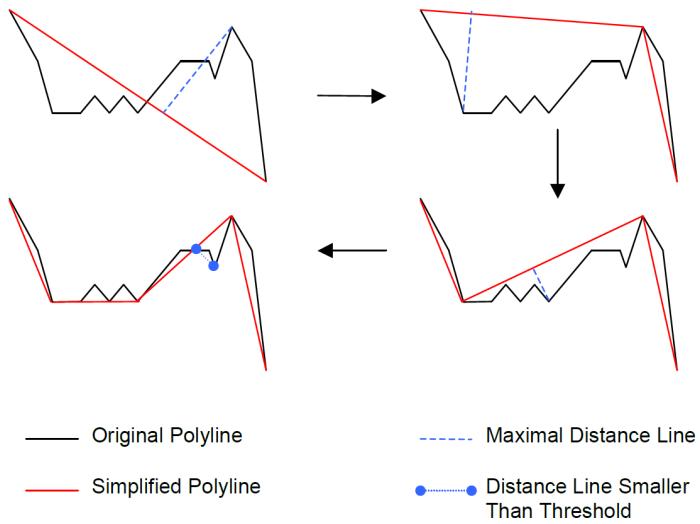
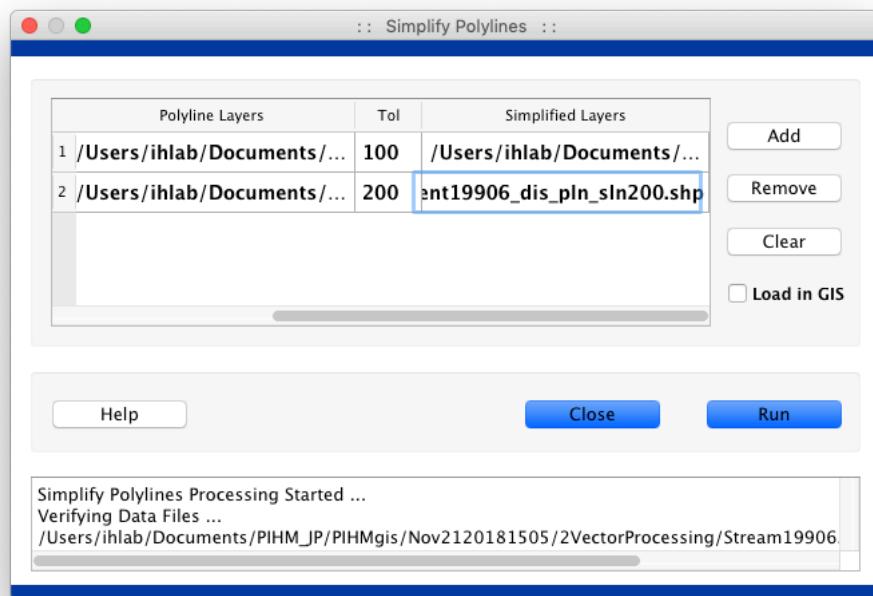


Figure 6.4 Algorithm for Simplification

290
291

292



293
294

Figure 6.5 Simplify Polyline dialog

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

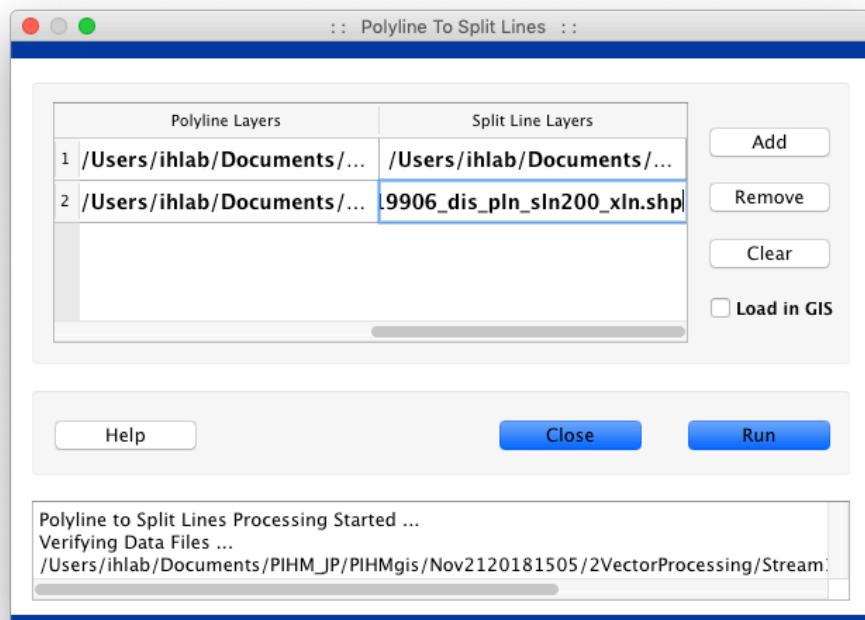
299 **6.4 Polyline to Split Lines and modifications**

300 For creating triangle elements in 'Domain Decomposition', all line
301 segments should be considered as an individual feature object. 'Split
302 Lines' module splits polylines at each vertex. 'Split Lines' module is
303 described schematically in Fig. 6.6. Input has only one feature which is
304 of type polyline. Output has three features which are of the type lines.
305 The name of output file has "_xln" at the end to distinguish easily.
306



307
308 *Figure 6.6 Schematically described split line operation*

309

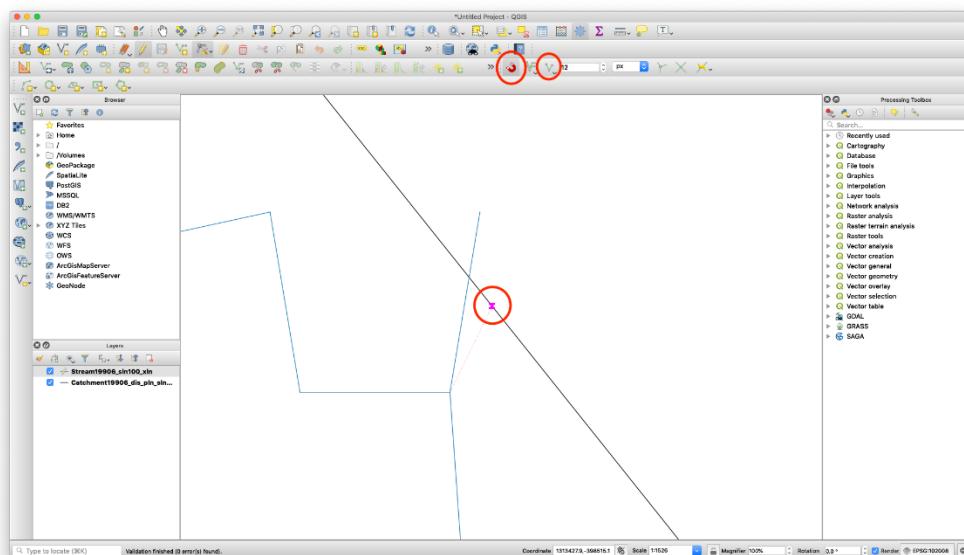


310
311 *Figure 6.7 Polylines To Split Lines dialog*

312 Before moving on to the next step, the result of 'Split Lines' should be
313 checked. In Fig. 6.8, you may see the stream outlet (blue line) and
314 catchment boundary (black line) don't meet each other (it is also
315 possible that sometimes the stream may intersect the catchment
316 boundary but not end at it). This problem is caused by simplification. To
317 address this problem, 'editing' tool should be used to move a nearby
318 catchment boundary vertex to the river outlet. The end point of stream
319 outlet could be moved to the catchment boundary vertex, but in this
320 case you have to make sure that the elevation at the moved location is
321 lower than at the upstream stream vertex.
322

Fig. 6.8 shown the editing process in QGIS.

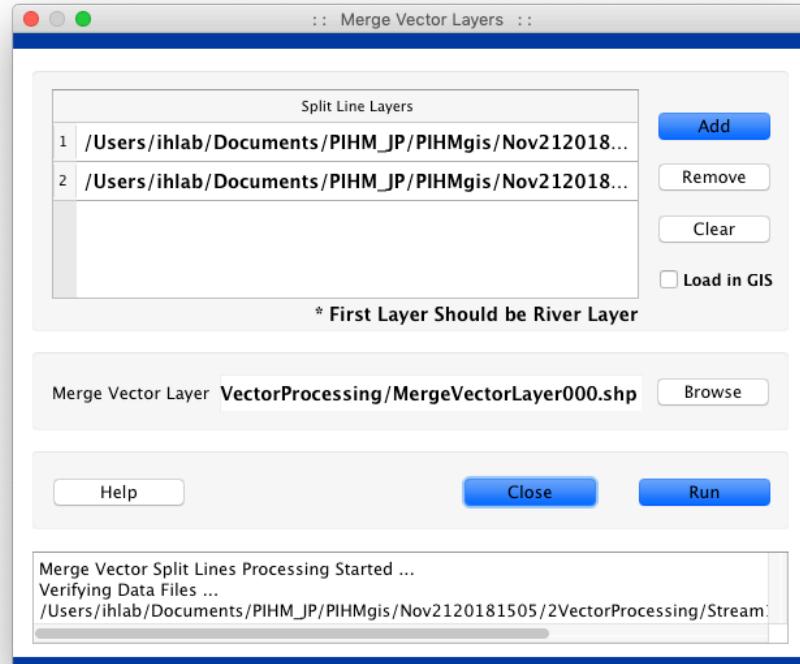
323



324
325

Figure 6.8 Modifying split stream and catchment boundary with snapping option

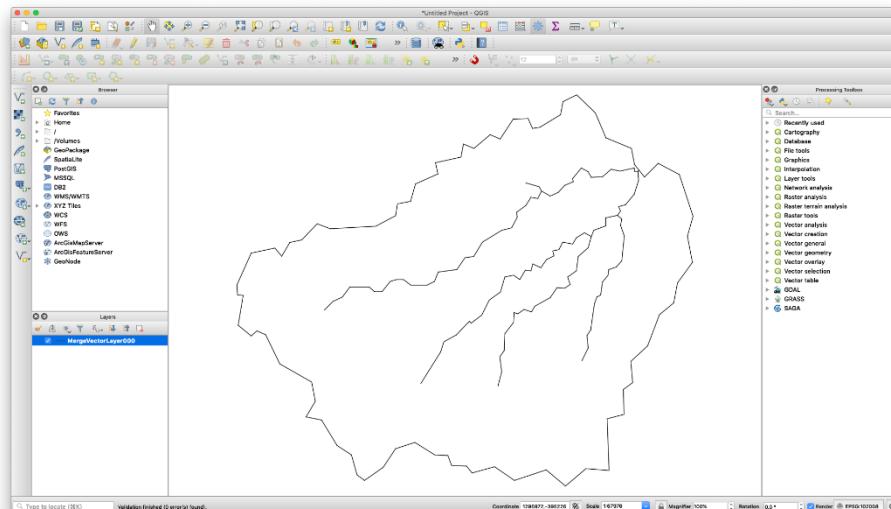
326
327 **6.5 Vector Merge**
328 'Vector Merge' is the final step of 'Vector Processing'. It merges stream
329 and catchment boundary (and other internal boundaries if they exist),
330 so that a merged shape file will be used in 'Domain Decomposition'. Fig.
331 Fig. 6.10 shows the result of 'Vector Processing'.
332



333
334

Figure 6.9 Merge Vector Layers dialog

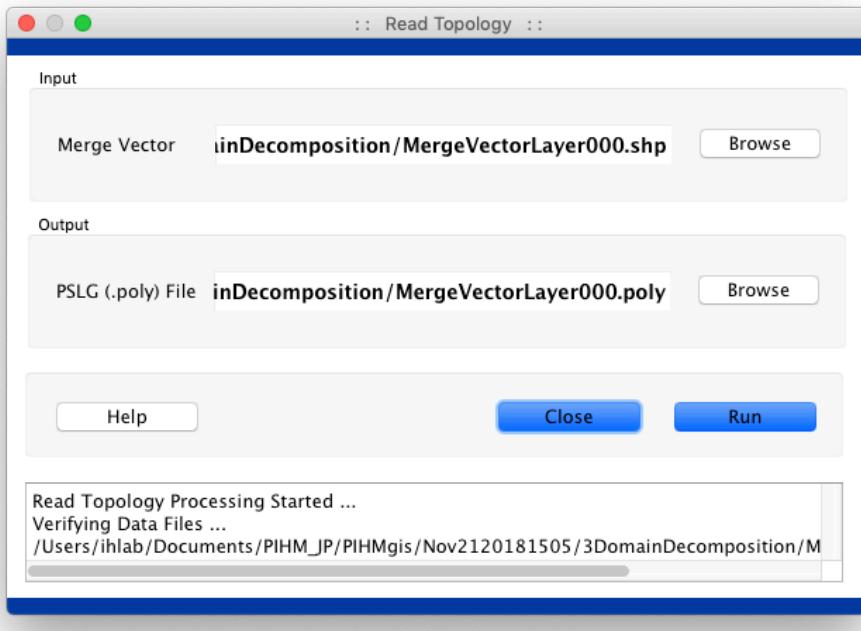
335



336
337

Figure 6.10 Split lines including stream line and catchment boundary

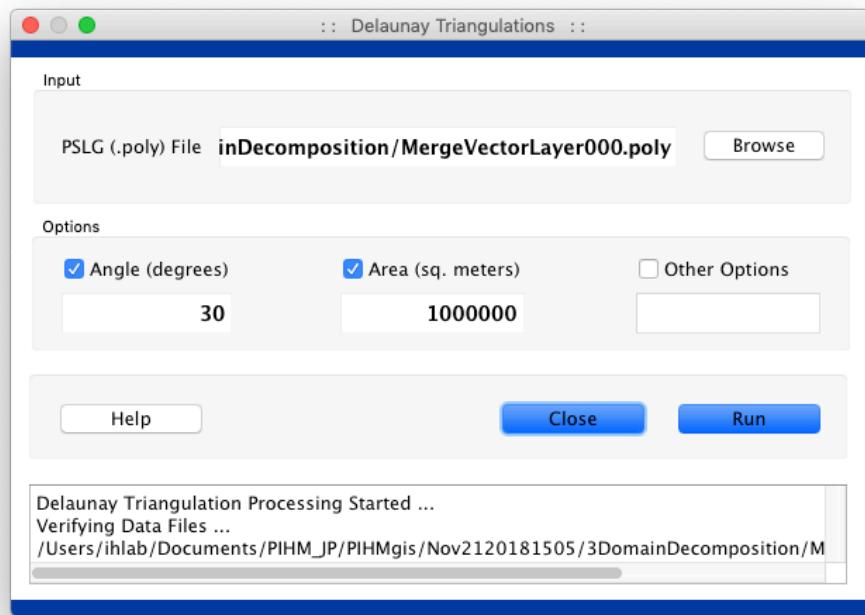
338 **7 Domain Decomposition**
339 'Domain Decomposition' is the process of creating mesh using internal
340 and external boundary split lines (of stream, catchment boundary etc.).
341 This is done using the following steps:
342 **7.1 Read Topology**
343 All the vector line information from the merged shape file is extracted
344 into planer straight line graph (PSLG), and is saved in a '.poly' file.



345
346 *Figure 7.1 Read Topology dialog*

347 **7.2 Triangulation**
348 The library 'TRIANGLE' developed by Shewchuk (2001) is one of the
349 most efficient implementations of constrained Delaunay triangulation
350 algorithm which provides numerous flexibility to the user regarding the
351 number and distribution of triangles. It takes planner straight line graph
352 (PSLG) as input. The algorithm works in such a way that it refines the
353 Delaunay triangulation by inserting carefully placed vertices until the
354 generated mesh meets a provided quality and size criterion.
355 'Angle (minimum angle)' and 'Area (maximum area)' constraint options
356 are provided in PIHMGis interface. You may use 'Others' option to access
357 other TRIANGLE program commands. These options are provided in
358 Table 7.1. This step generates '.poly', '.node', '.ele', and '.neigh' files,
359 which define the mesh.

360 In this tutorial, min. angle was set as 30 degree (to avoid sharp
361 angle problem) and max. area was set as 1,000,000 m² to create small
362 number of mesh grids to shorten the run time.
363



364
365

Figure 7.2 TRIANGULATION dialog

366
367

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

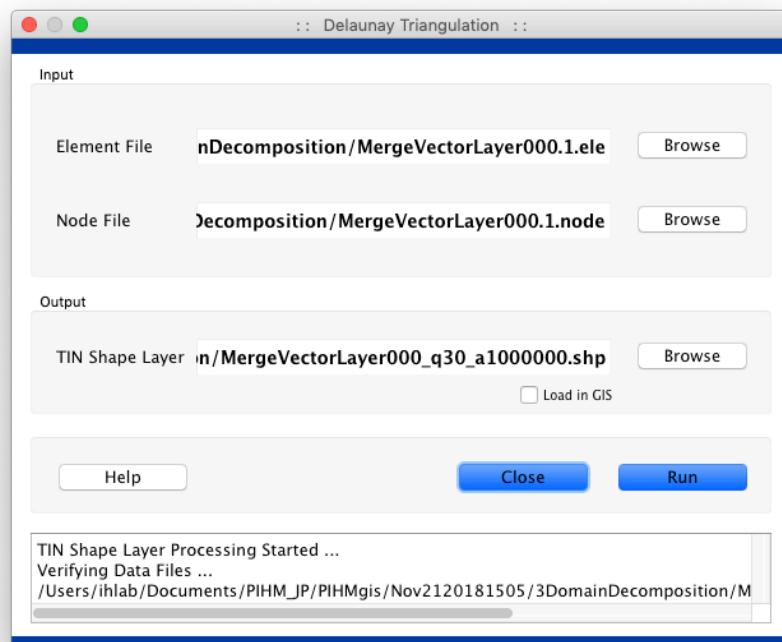
Options	Description
-p	Triangulates a Planar Straight Line Graph (.poly file).
-r	Refines a previously generated mesh.
-q	Quality mesh generation. A minimum angle may be specified.
-a	Applies a maximum triangle area constraint.
-u	Applies a user-defined triangle constraint
-A	Applies attributes to identify 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.

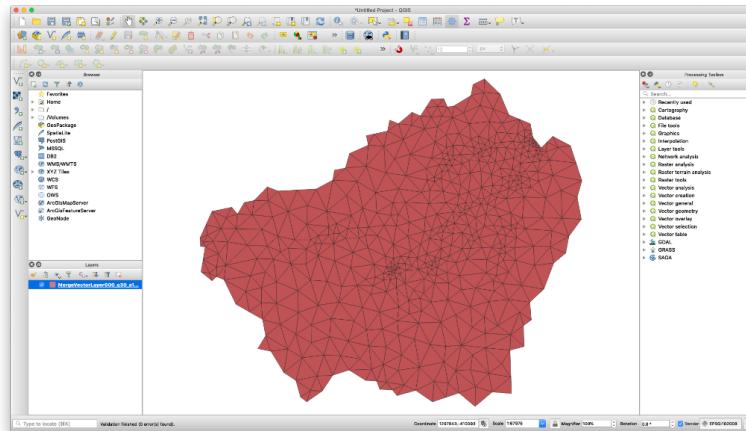
368

369 **7.3 Tin Shape Layer**370 In this step, '.ele' file and '.node' files are used to generate TIN shape
371 file as shown in Fig. 7.4. Target watershed was discretized into 969
372 triangles.

373

374
375*Figure 7.3 Tin Shape Layer dialog*

376



377
378

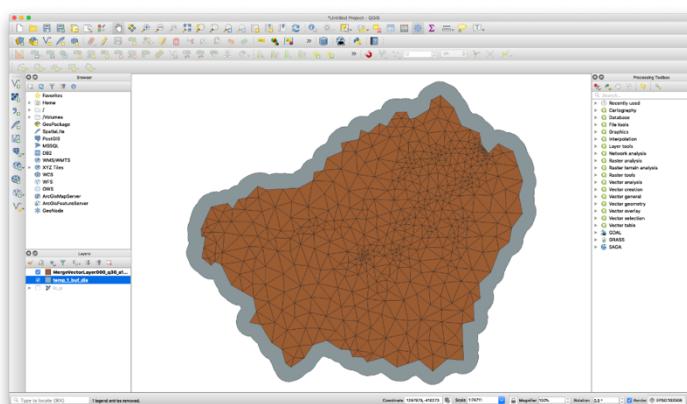
Figure 7.4 Created triangle mesh grids

379 8 Pre-processing raw data

380 8.1 Introduction

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

390 It is imperative that the raw data used here fully covers the
391 watershed. Users are advised to use a buffer shape file that is larger by
392 around 2 times the size of the coarsest grid . As shown in Fig. 8.1, buffer
393 shape file is large enough to cover mesh shape file. This buffer shape
394 file will be used to clip raster input layers.
395



396
397
398

Figure 8.1 Mesh shape file and buffer shape file

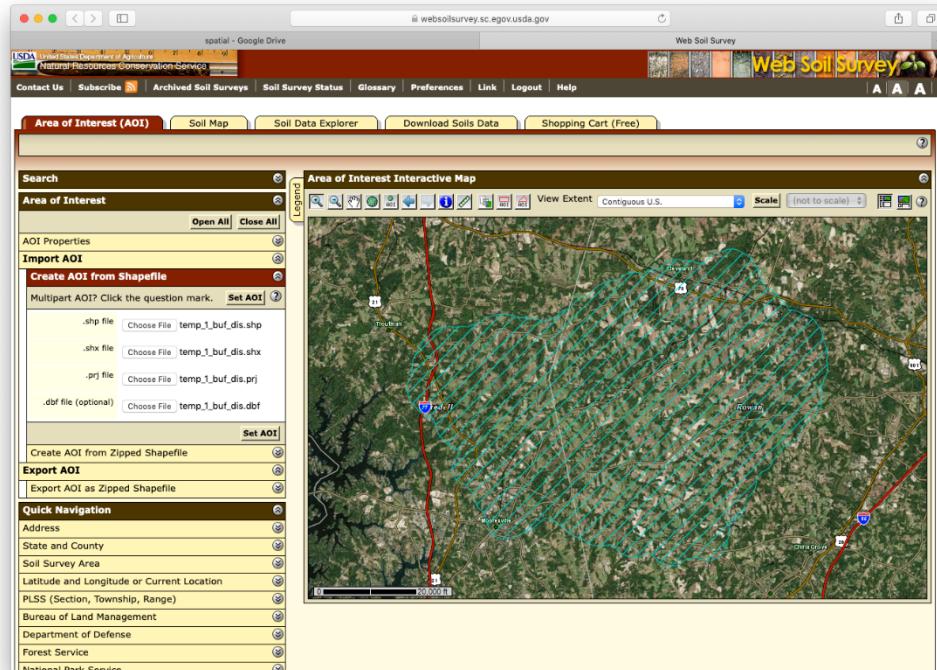
399 **8.2 Soil Data**

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

403 Easiest way to download and extract soil attributed data
404 (especially SSURGO) is using 'Soil Data Viewer⁸' that is a plug-in
405 application of ArcGIS, and 'Soil Data Development Toolbox⁹' that is a
406 library of ArcGIS. The 'Soil Data Viewer' links shape file and database of
407 SSURGO to extract relevant soil parameters. The 'Soil Data
408 Development Toolbox' includes downloading, mapping, data
409 management, and reporting tools. Even though this tutorial focused on
410 using QGIS in other sections as it is open-source, ArcGIS is used in this
411 section. If ArcGIS is not available, you should extract parameters from
412 database on your own.

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

416



417
418

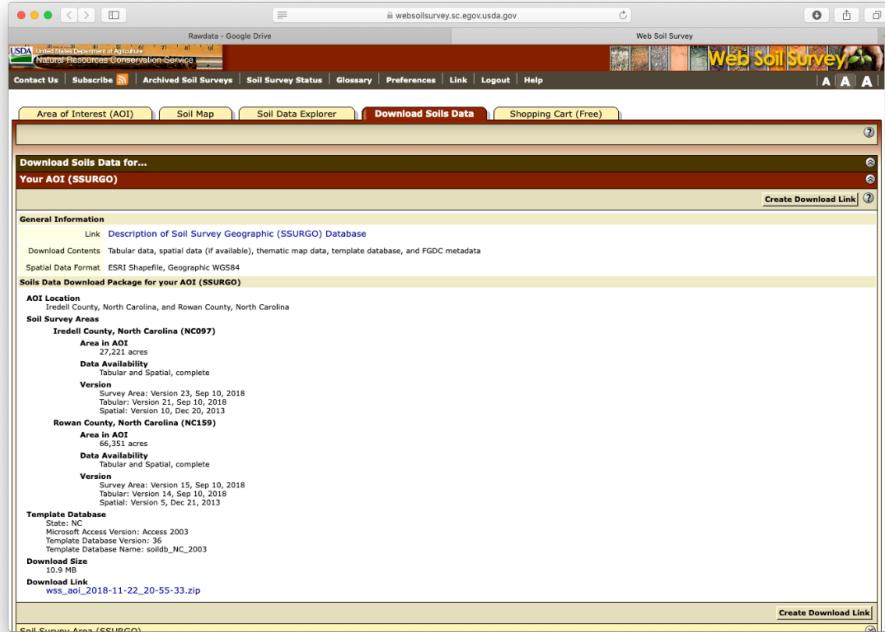
Figure 8.2 'Web Soil Survey' website and AOI of buffer shape file

419 You can download soil data as shown in Fig. 8.3. After selecting AOI,
420 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

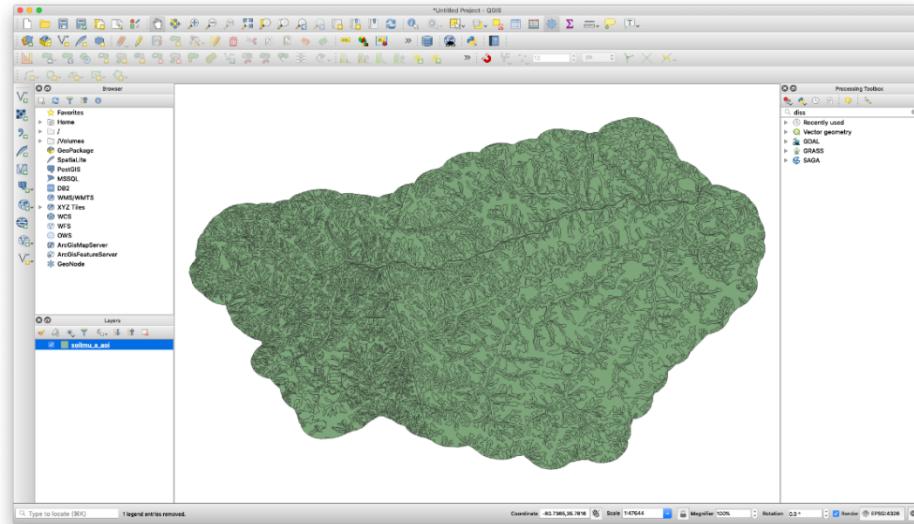
⁹ https://www.nrcs.usda.gov/wps/portal/nrcs/detail/soils/survey/geo/?cid=nrcs142p2_053628

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



425
 426 *Figure 8.3 Download link of soil data shape file on the WSS website*

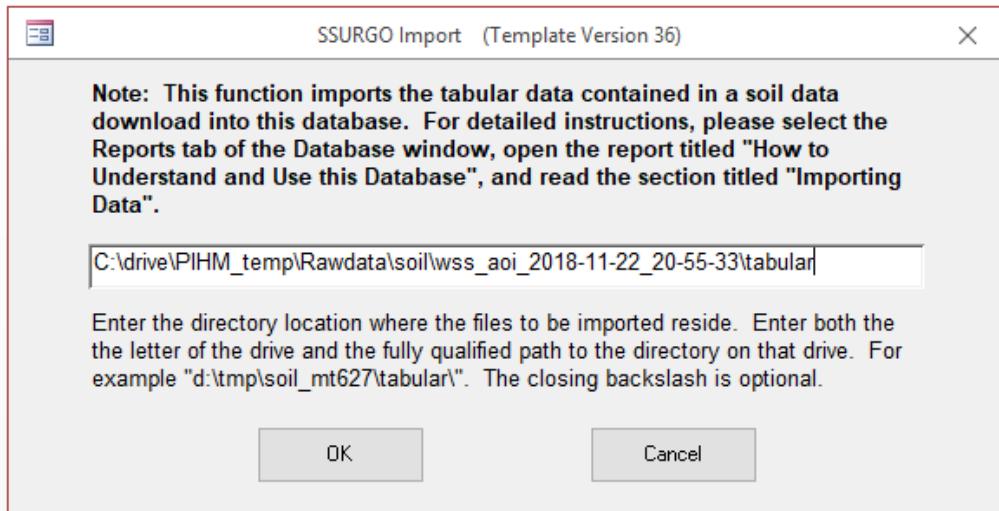
427



428
 429 *Figure 8.4 SSURGO shape file for target watershed*

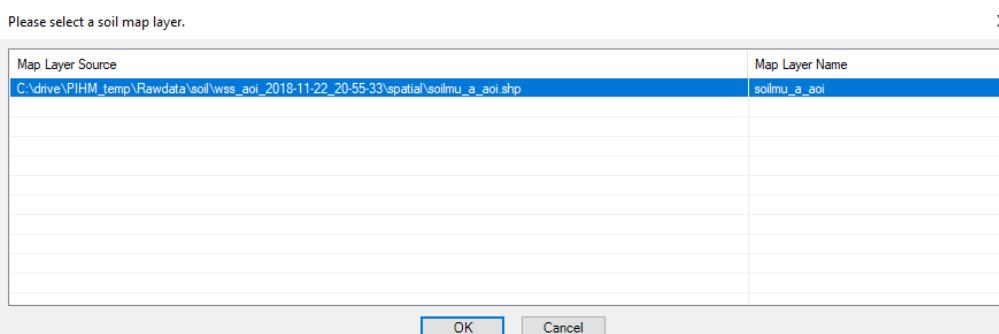
430 For generating soil (.soil) and geology (.geol) input data in sections 9.5
 431 and 9.6, four parameters for each MUKEY should be extracted from the
 432 SSURGO database. In this regard, first open the MS Access database

433 file ('.mdb') that is included in the compressed file downloaded from the
434 WSS website. When the Access database file is opened the first time, a
435 dialog shown in Fig. 8.5 appears. Fill the path of 'tabular' folder in the
436 blank space. It allows Soil Data Viewer to use SSURGO database.
437



438
439 *Figure 8.5 SSURGO import dialog for opening '.mdb' database*

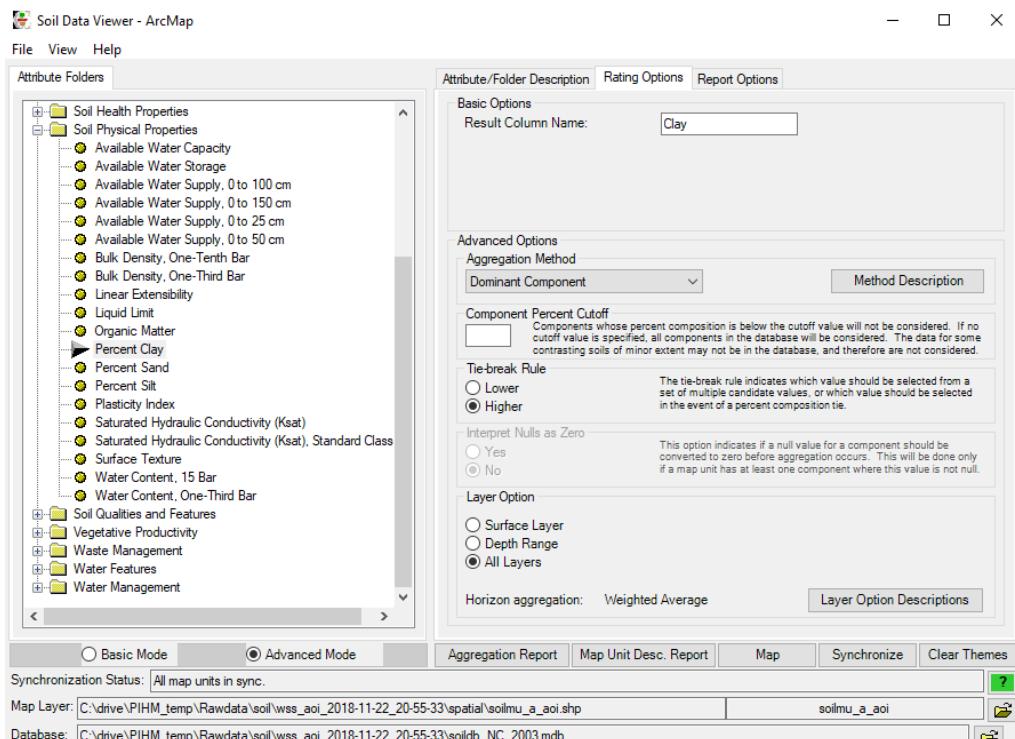
440
441 While opening AOI shape file, when you turn on the Soil Data Viewer,
442 you will see a dialog shown in Fig. 8.6. Note that, you should use the
443 downloaded shape file here with no modifications. Modified or pre-
444 processed shape file will generate an error.
445



446
447 *Figure 8.6 Dialog for selecting soil map layer for using Soil Data Viewer*

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

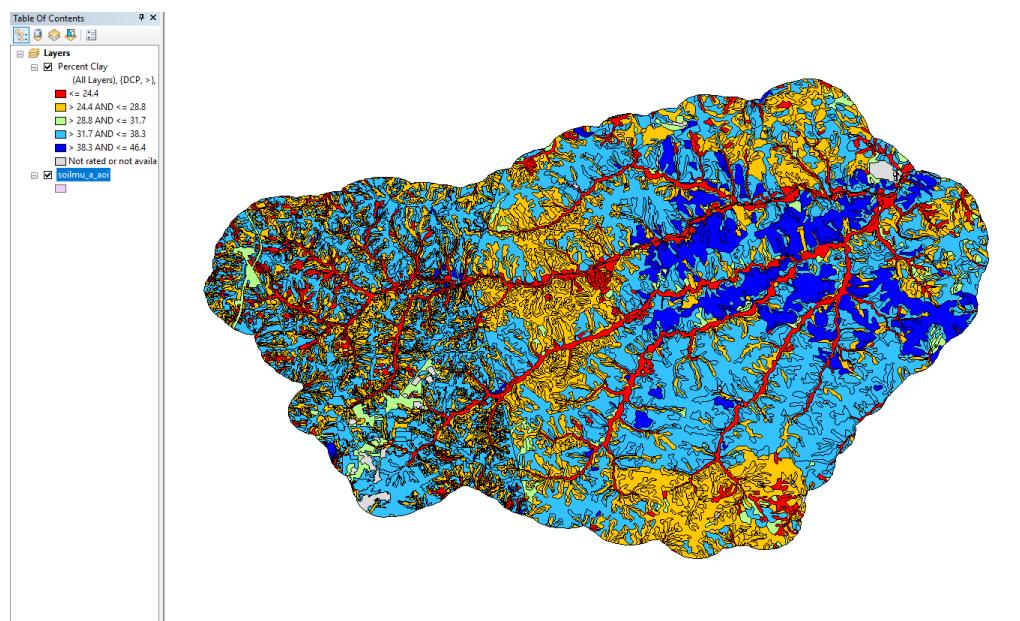
456 using Soil Data Viewer. You should specify several options in the dialog
 457 of Fig. 8.7 while extracting the layer in Fig. 8.8. Most important is 'Layer
 458 Option' in the 'Rating Options' tab. You can specify exact depth range,
 459 or choose surface layer, but this tutorial chose all layers.
 460



461
462

Figure 8.7 Soil Data Viewer dialog

463

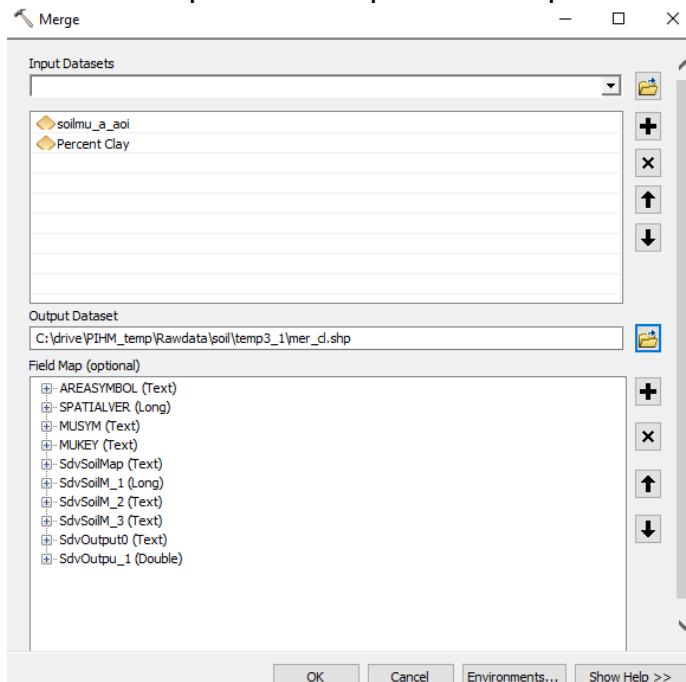


464
465

Figure 8.8 Clay percent extracted from SSURGO using Soil Data Viewer

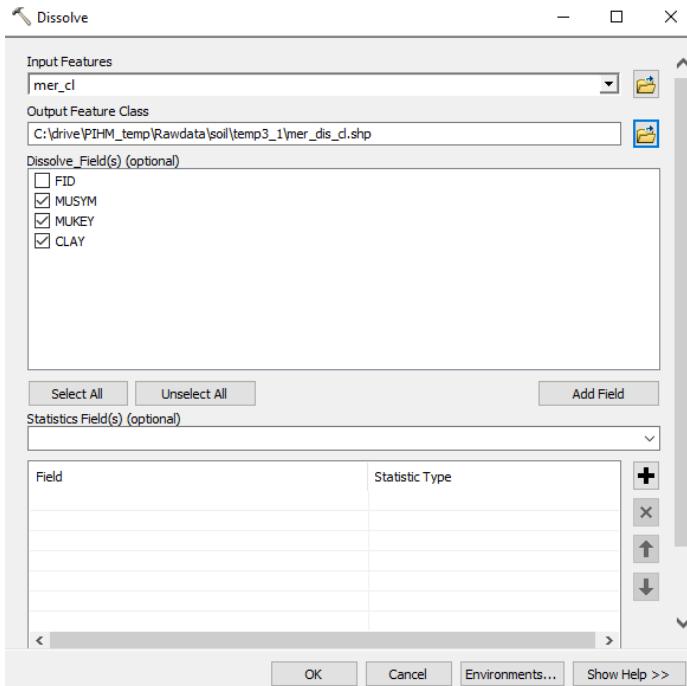
466

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



474

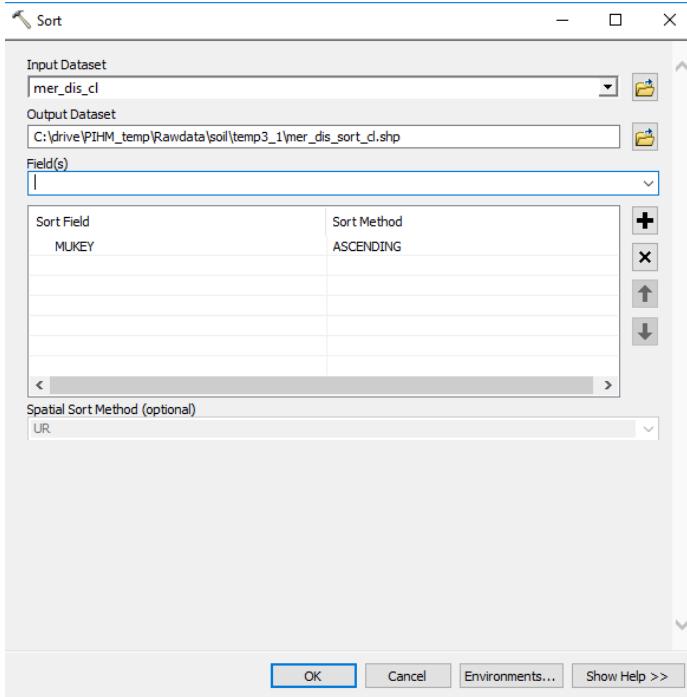
Figure 8.9 Merging dialog of ArcMap for clay percent of each feature



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

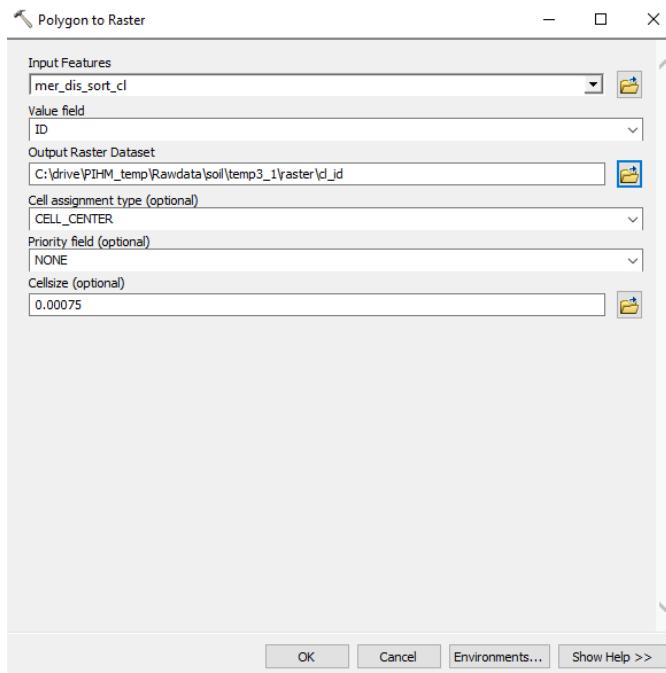
481

482
483
484

FID	Shape *	MUSYM	MUKEY	CLAY	ID
0	Polygon	AgB	114961	33.5	1
1	Polygon	CdA	114970	32.3	2
2	Polygon	CdB	114971	36.7	3
3	Polygon	CdC	114972	33.2	4
4	Polygon	ChA	114976	29.9	5
5	Polygon	ChB	114977	31.4	6
6	Polygon	DsB	114979	32.8	7
7	Polygon	ErB	114980	28	8
8	Polygon	EnC	114981	28	9
9	Polygon	ExC	114982	26.0	10
10	Polygon	ExD	114983	28.6	11
11	Polygon	EuB	114984	28	12
12	Polygon	HeB	114985	28.8	13
13	Polygon	HeC	114986	34.1	14
14	Polygon	MnB	114992	30.2	15
15	Polygon	MnB	114993	30.2	16
16	Polygon	PdB	114998	26.2	17
17	Polygon	PdB	114999	26.2	18
18	Polygon	PeD	115000	27.2	19
19	Polygon	PeE	115001	26.2	20
20	Polygon	RdC	115010	16.6	21
21	Polygon	RdC	115011	16.6	22
22	Polygon	SeA	115014	27.7	23
23	Polygon	SeB	115015	30.2	24
24	Polygon	SeC	115016	30.2	25
25	Polygon	SeB	115018	34	26
26	Polygon	Uf	115023	26	27
27	Polygon	Uf	115024	0	28
28	Polygon	UhC	115025	35.4	29
29	Polygon	UhC	115026	35.4	30
30	Polygon	UhB	115027	29.3	31
31	Polygon	VAC	115031	29	32

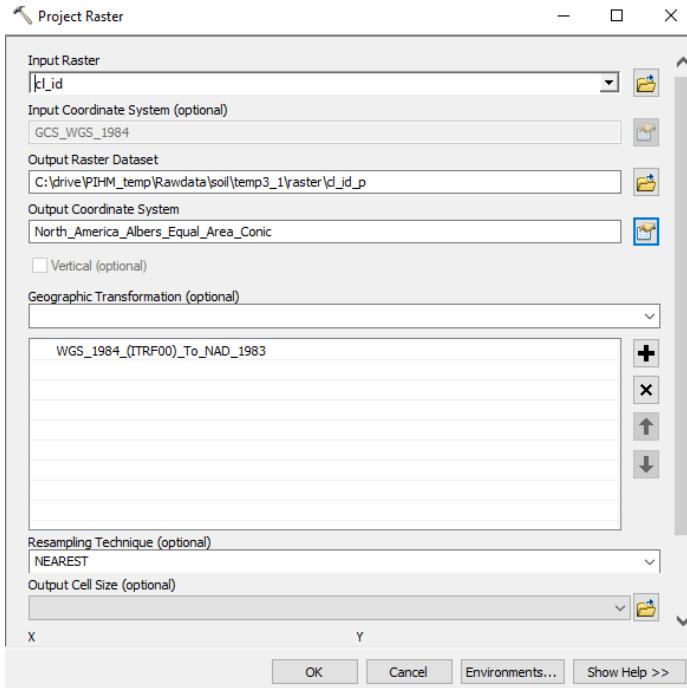
Figure 8.12 Attribute table of clay percent shape file after merging, dissolving, sorting, and inserting ID column

485 For creating '.att' file and allocating soil types to each triangular
486 element, you need the '.asc' raster file with id number that indicates
487 each soil type. First, you can make raster file based on the ID column
488 as shown in Fig. 8.13. As PIHM needs spatial data to be in the projected
489 coordinate system, appropriate projection transformations should be
490 performed (Fig. 8.14). Finally, as shown in Fig. 8.15, '.asc' file should
491 be generated, because PIHMGis requires '.asc' format to read the raster.
492 This '.asc' raster file based on id will be used in section 9.3.
493



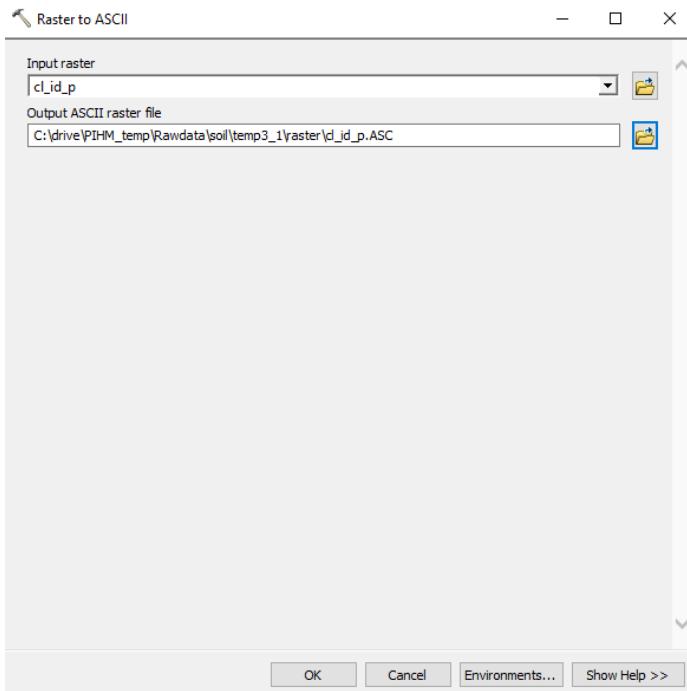
494
495

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



496
497

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



498
499

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

500
501 You may need several soil parameters. For example, PIHM needs clay
502 percent, silt percent, organic matter, and bulk density. Shape files
503 similar to one shown in Fig. 8.12 for each soil parameter can be
504 generated. It is to be noted that there may be some grids where data

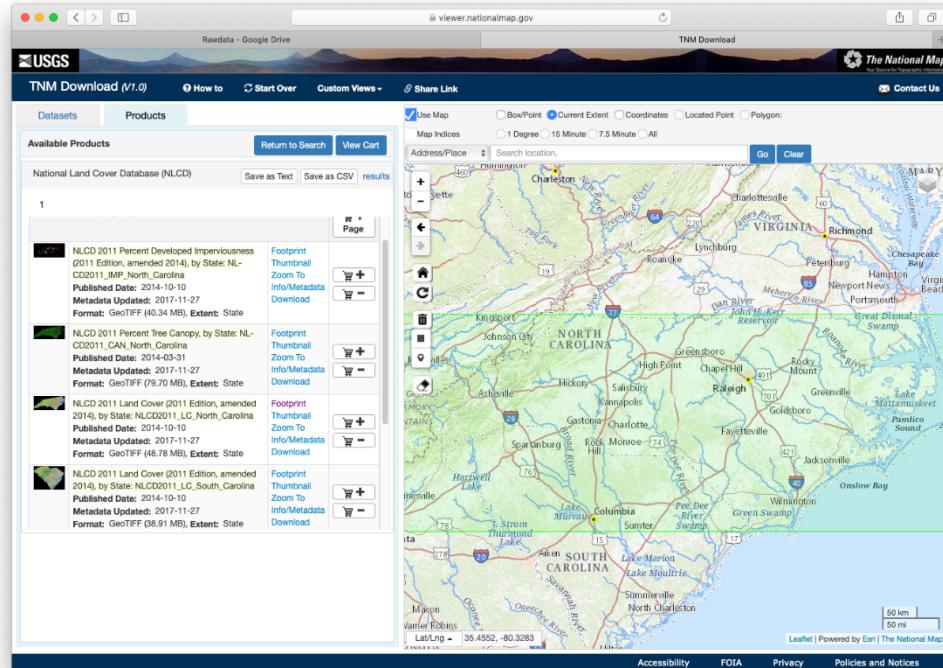
505 does not exist. These could be locations covered by water perennially.
506 For these grids, an option is use majority filter to fill in the parameter
507 values.

508 8.3 Land Cover Data

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

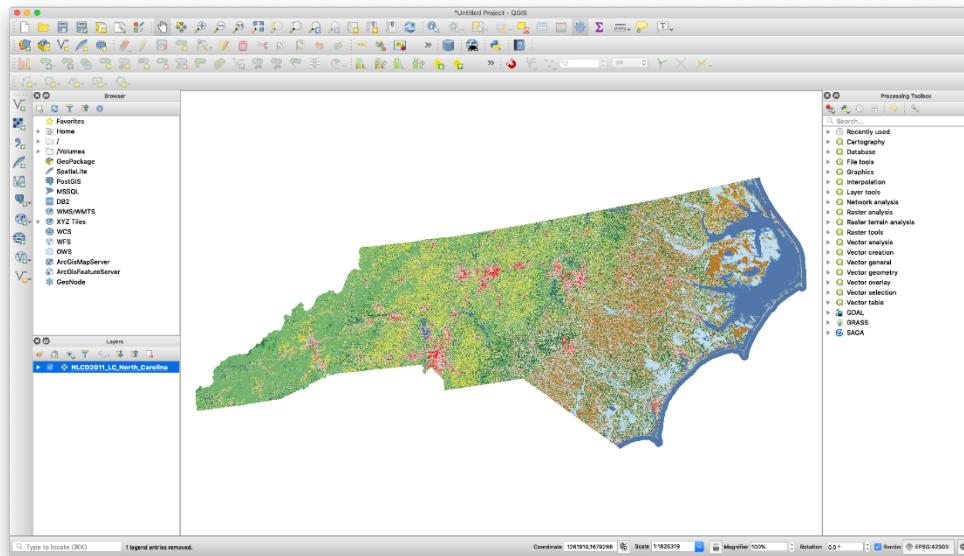
512 Land cover data can also be downloaded at the TNM Download
513 v1.0 (Fig. 8.16). This tutorial used NLCD 2011 for North Carolina (Fig.
514 8.17).

515



516
517

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



518
519

Figure 8.17 NLCD 2011 for North Carolina

520 Before moving forward, have a quick look at Figs. 8.18 and 8.19. Fig.
521 Fig. 8.18 shows the legend of NLCD 2011. Fig. 8.19 shows the legend of
522 UMD Land Cover Classification. Since PIHMgis generates PIHM land
523 cover parameter file (.lc) file based on NLCD, you will have to
524 accordingly reclassify if you are using other land cover data such as the
525 UMD Land Cover Classification (UMDLCC).
526

Class Value	Classification Description	
Water	11Open Water- areas of open water, generally with less than 25% cover of vegetation or soil.	Shrubland
	12Perennial Ice/Snow- areas characterized by a perennial cover of ice and/or snow, generally greater than 25% of total cover.	51Dwarf 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.
Developed	21Developed, 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.	52Shrub/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.
	22Developed, Low Intensity- areas with a mixture of constructed materials and vegetation. Impervious surfaces account for 20% - 49% percent of total cover. These areas most commonly include single-family housing units.	71Grassland/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.
	23Developed, 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.	72Edge/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 tufted tundra.
	24Developed 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.	73Lichens- Alaska only areas dominated by fruticose or foliose lichens generally greater than 80% of total vegetation.
Barren	31Barren Land (Rock/Sand/Clay) - areas of bedrock, desert pavement, scarp, 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.	74Moss- Alaska only areas dominated by mosses, generally greater than 80% of total vegetation.
Forest	41Deciduous 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.	Planted/Cultivated
	42Evergreen 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.	81Pasture/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.
	43Mixed 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.	82Cultivated 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
		90Woody 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.
		95Emergent 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.

527
528
529

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

530

UMD Land Cover Classification				
Data Access				
<ul style="list-style-type: none"> Download via Search and Preview Tool (ESDI) Download via web page with links to FTP Server 				
Overview				
<p>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.</p>				
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

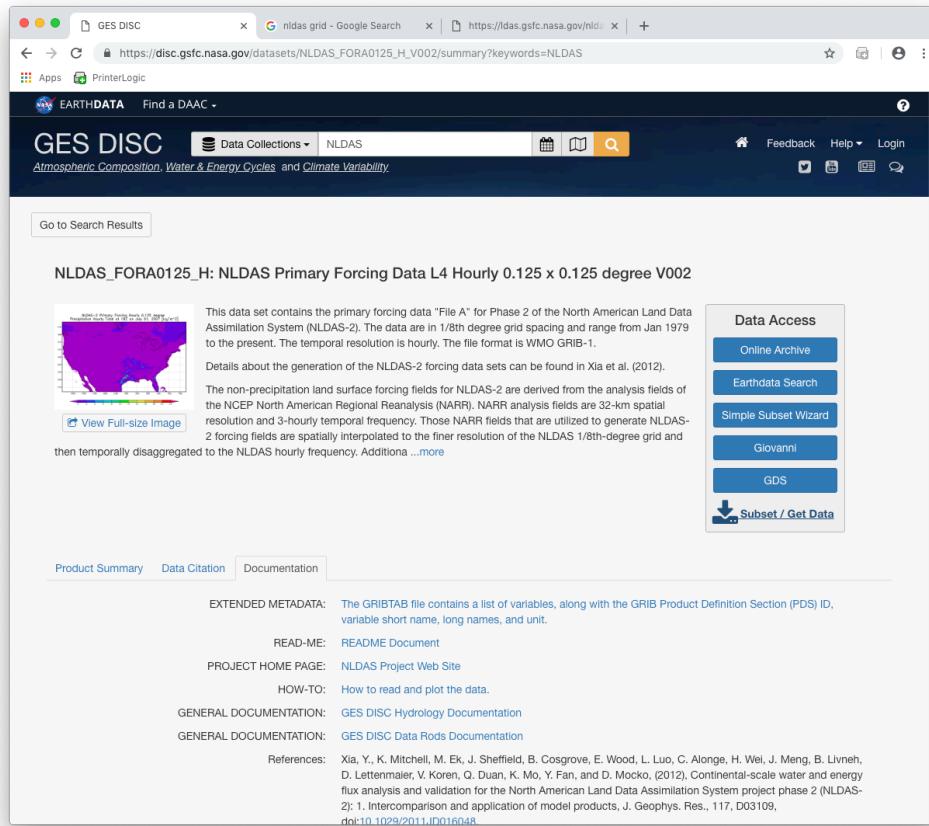
Figure 8.19 Legend of UMD Land Cover Classification (<http://glcf.umd.edu/data/landcover/>)

531
532
533
534

535 8.4 Forcing Data

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

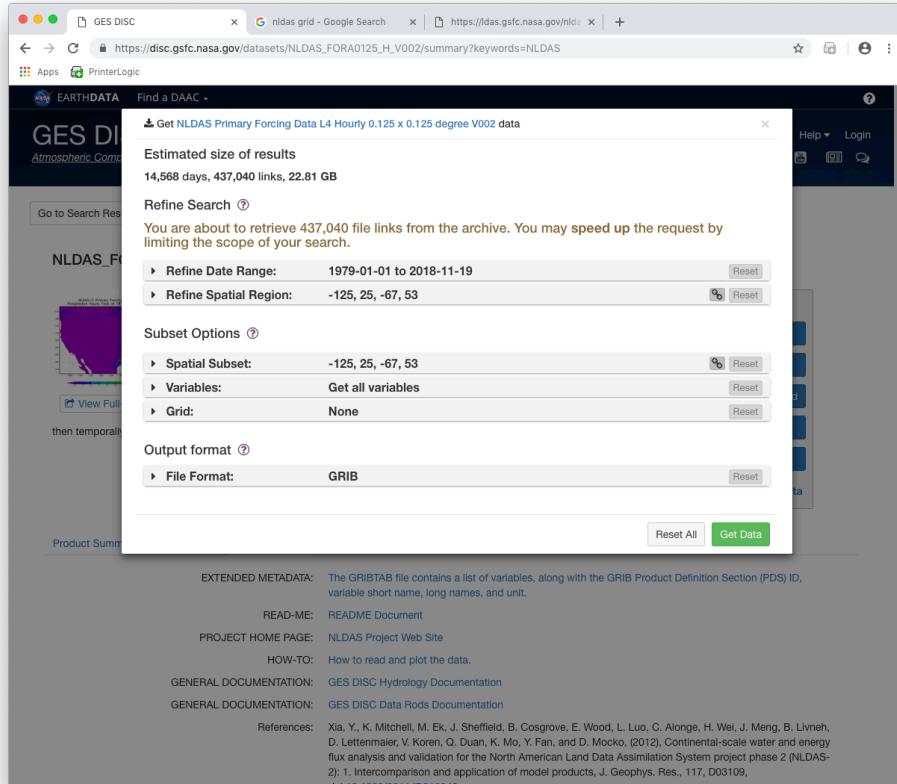
540 In this tutorial, NLDAS 2 forcing data is used. The official full name
541 is 'NLDAS Primary Forcing Data L4 Hourly 0.125 x 0.125 degree V002'.
542 Users can use other climate or weather data for input. More information
543 about the NLDAS-2 product is available in README document at
544 <https://hydro1.gesdisc.eosdis.nasa.gov/data/NLDAS/README.NLDAS2.pdf>. Fig.
545 8.20 shows NASA GES DISC website
546 (https://disc.gsfc.nasa.gov/datasets/NLDAS_FORA0125_H_V002/summary) that
547 users can download NLDAS data from. In 'Data Access', there are
548 several ways to download serial climate data. This tutorial used linux
549 server, but users can use any other method explained in website
550 (<https://disc.gsfc.nasa.gov/data-access>).
551



552
553

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

554 After registering an Earthdata account on GES DISC website, when you
 555 click 'Subset / Get Data' in Fig. 8.20, you will see a pop-up as in Fig.
 556 8.21. After specifying data range and format on this pop-up and
 557 pressing 'Get Data' button, you will get a text file that contains download
 558 links. Now, you can download NLDAS-2 files on your computer as shown
 559 in Fig. 8.22.
 560



561
562

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

563

```
ihlab — jpark102@uahpc:~ — ssh ualjxp@dmc.asc.edu — 121x13
#!/bin/bash
#SBATCH --qos large
#SBATCH --job-name=nldas_7983
#SBATCH --output=/home/ualjxp/NLDASdownload_7983.log
#SBATCH --ntasks=1
#SBATCH --mem=4048
#SBATCH --time=24:00:00
cd /home/ualjxp/NLDAS/back/
wget --load-cookies ~/.urs_cookies --save-cookies ~/.urs_cookies --auth-no-challenge-on --keep-session-cookies --content-disposition -i /home/ualjxp/subset_NLDAS_FORA0125_H_V002_19790101_19831231.txt
~
```

564
565

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

566

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

```

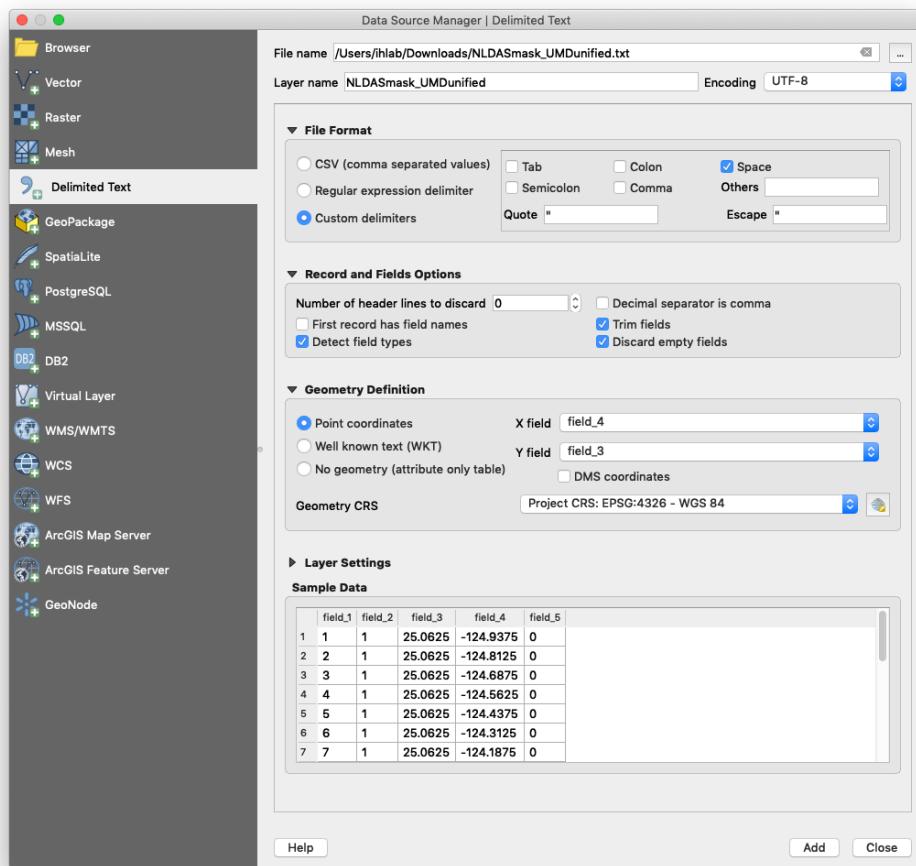
1 1 25.0625 -124.9375 0
2 1 25.0625 -124.8125 0
3 1 25.0625 -124.6875 0
4 1 25.0625 -124.5625 0
5 1 25.0625 -124.4375 0
6 1 25.0625 -124.3125 0
7 1 25.0625 -124.1875 0
8 1 25.0625 -124.0625 0
9 1 25.0625 -123.9375 0
10 1 25.0625 -123.8125 0
11 1 25.0625 -123.6875 0
12 1 25.0625 -123.5625 0
13 1 25.0625 -123.4375 0
14 1 25.0625 -123.3125 0
15 1 25.0625 -123.1875 0
16 1 25.0625 -123.0625 0
17 1 25.0625 -122.9375 0
18 1 25.0625 -122.8125 0
19 1 25.0625 -122.6875 0
20 1 25.0625 -122.5625 0
21 1 25.0625 -122.4375 0
22 1 25.0625 -122.3125 0
23 1 25.0625 -122.1875 0
24 1 25.0625 -122.0625 0
25 1 25.0625 -121.9375 0
26 1 25.0625 -121.8125 0
27 1 25.0625 -121.6875 0
28 1 25.0625 -121.5625 0
29 1 25.0625 -121.4375 0
30 1 25.0625 -121.3125 0
31 1 25.0625 -121.1875 0
32 1 25.0625 -121.0625 0
33 1 25.0625 -120.9375 0
34 1 25.0625 -120.8125 0
35 1 25.0625 -120.6875 0
36 1 25.0625 -120.5625 0
37 1 25.0625 -120.4375 0
38 1 25.0625 -120.3125 0
39 1 25.0625 -120.1875 0
40 1 25.0625 -120.0625 0
41 1 25.0625 -119.9375 0
42 1 25.0625 -119.8125 0
43 1 25.0625 -119.6875 0
44 1 25.0625 -119.5625 0
45 1 25.0625 -119.4375 0
46 1 25.0625 -119.3125 0
47 1 25.0625 -119.1875 0
48 1 25.0625 -119.0625 0
49 1 25.0625 -118.9375 0
50 1 25.0625 -118.8125 0
51 1 25.0625 -118.6875 0
52 1 25.0625 -118.5625 0
53 1 25.0625 -118.4375 0
54 1 25.0625 -118.3125 0
55 1 25.0625 -118.1875 0
56 1 25.0625 -118.0625 0
57 1 25.0625 -117.9375 0
58 1 25.0625 -117.8125 0
59 1 25.0625 -117.6875 0
60 1 25.0625 -117.5625 0
61 1 25.0625 -117.4375 0
62 1 25.0625 -117.3125 0

```

576
577
578

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

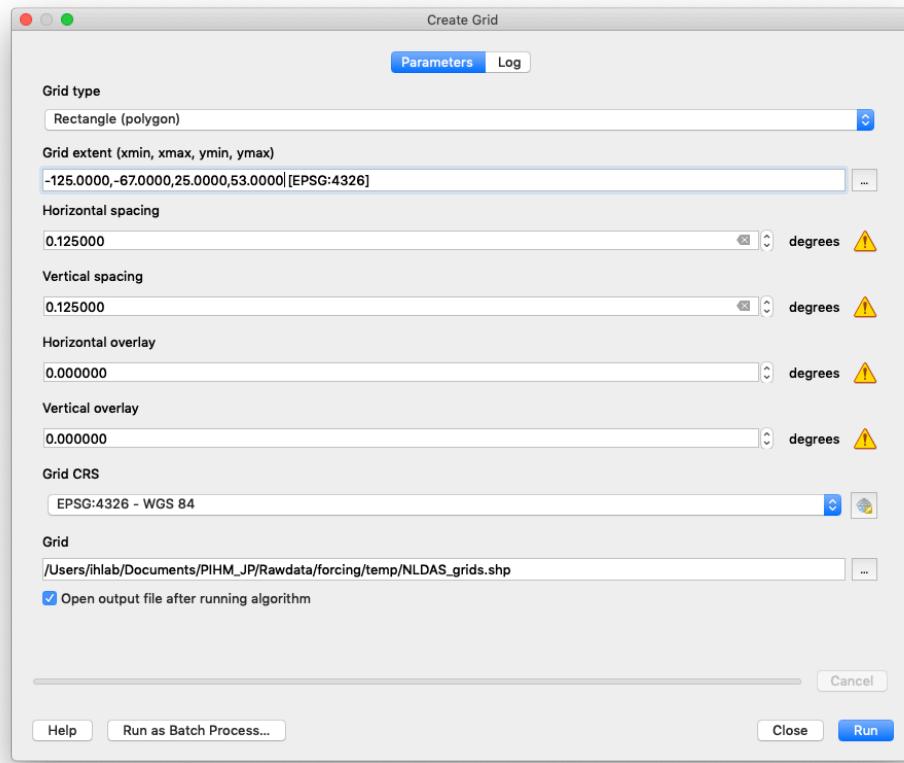
579



580
581

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

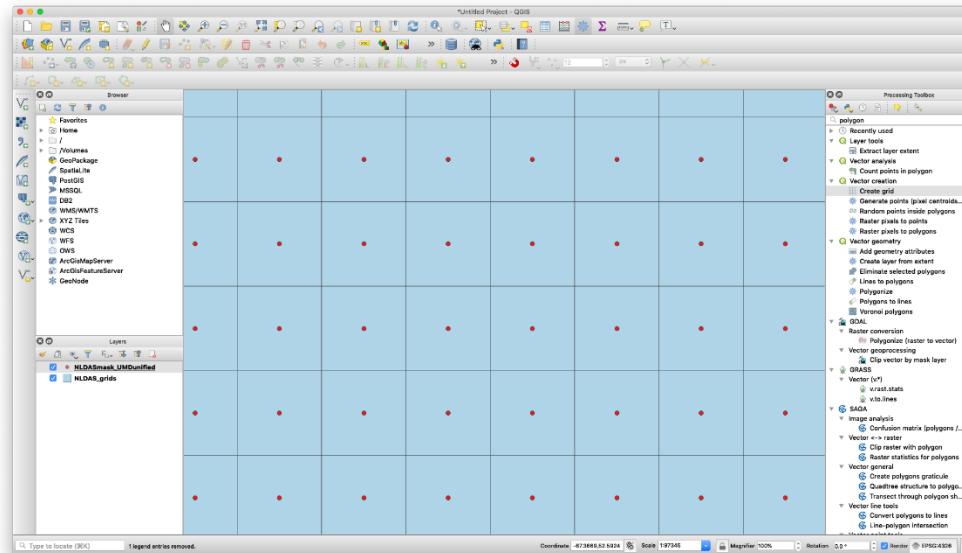
582



583
584
585

Figure 8.25 The dialog for generating rectangular polygons that resembles NLDAS-2 grid mesh

586

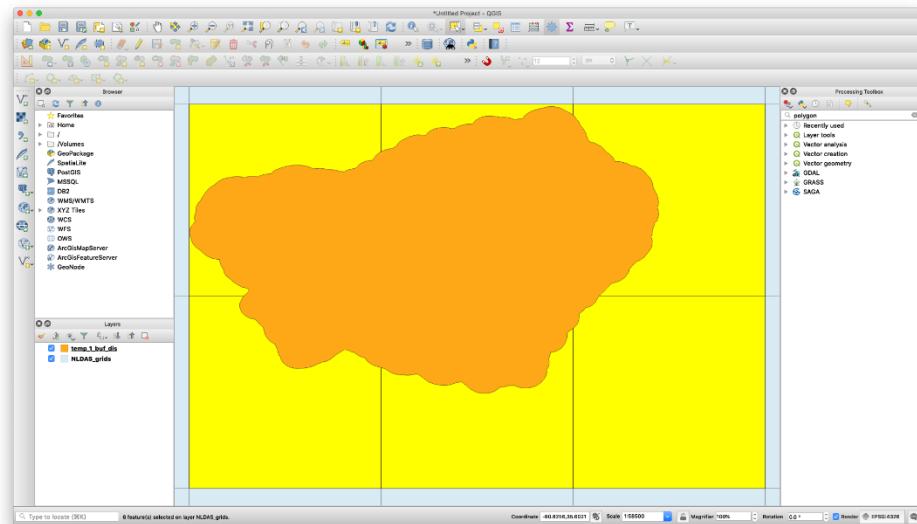


587
588
589

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

590

591 Using Fig. 8.26, we can generate spatial reference of NLDAS-2
 592 grid. Fig. 8.27 shows the buffered target watershed (orange
 593 color), selected NLDAS-2 grids that cover the target watershed
 594 (yellow), and other grids (blue). You need to export these selected
 595 polygons as a shape file and modify attribute table by adding new
 596 id number column as shown in Fig. 8.28. This integer will be used
 597 when you generate '.forc' file. This polygon shape file should be
 598 the '.asc' raster file (This translating procedure is not provided
 599 because it was repeated in this tutorial several times).
 600



601 *Figure 8.27 Buffered target watershed (orange) and selected NLDAS-2 grids (yellow)*
 602 *to cover target watershed*
 603

604

 A screenshot of the QGIS attribute table window titled 'grids_sel_p :: Features Total: 6, Filtered: 6, Selected: 0'. The table has columns: left, top, right, bottom, id, and ID_1. The data is as follows:

	left	top	right	bottom	id	ID_1
1	-80.875000...	35.6250000...	-80.750000...	35.500000...	79212	1
2	-80.750000...	35.6250000...	-80.625000...	35.500000...	79436	2
3	-80.625000...	35.6250000...	-80.500000...	35.500000...	79660	3
4	-80.875000...	35.7500000...	-80.750000...	35.625000...	79211	4
5	-80.750000...	35.7500000...	-80.625000...	35.625000...	79435	5
6	-80.625000...	35.7500000...	-80.500000...	35.625000...	79659	6

605 *Figure 8.28 Assign ID numbers for each grid to indicate grids using this ID*
 606

607 **9 DataModel Loder**

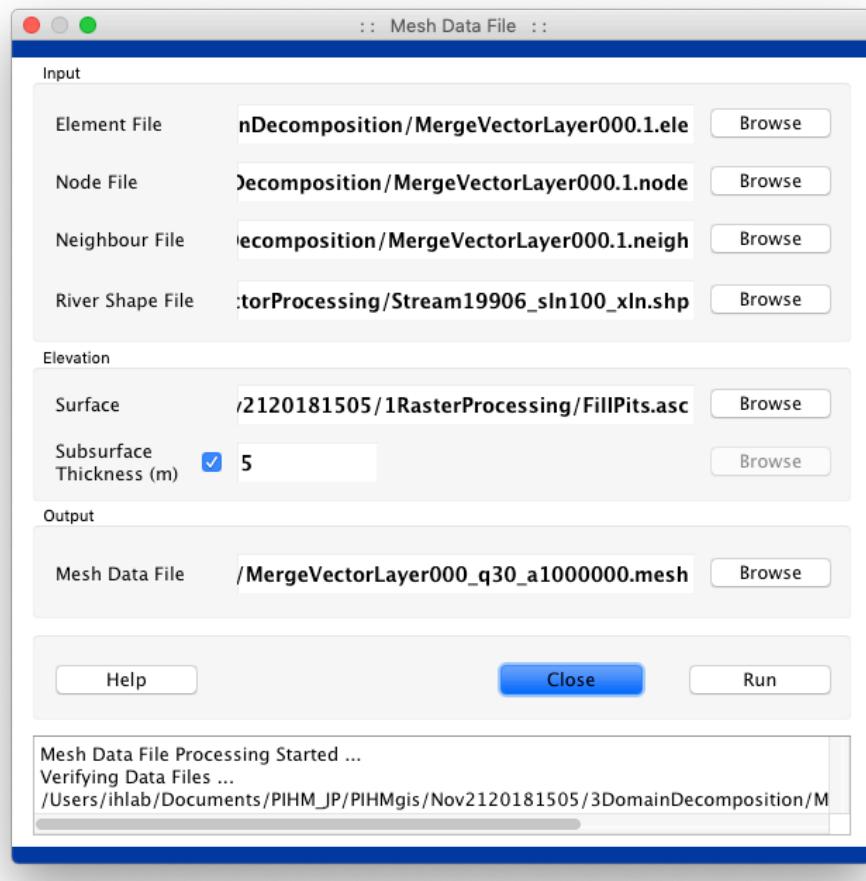
608 **9.1 Introduction**

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

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

619 **9.2 Mesh Data File**

620 Mesh file contains all spatial information of the triangular mesh. This
621 includes number of elements and nodes, neighbor elements for each
622 element, x and y coordinates of node, and elevation of each node are
623 included. The module generate '.mesh' file using previously generated
624 intermediate files. Fig. 9.1 shows the dialog box of this module. Table
625 9.1 shows the file structure of '.mesh', and Table 9.2 shows description
626 of variables included in the '.mesh' file.
627



628
629
630
631

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...					

632
633
634

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	1 st node of element	
Node[1]	Integer	2 nd node of element	
Node[2]	Integer	3 rd node of element	
Nabr[0]	Integer	1 st neighbor of element	0: boundary
Nabr[1]	Integer	2 nd neighbor of element	0: boundary
Nabr[2]	Integer	3 rd 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

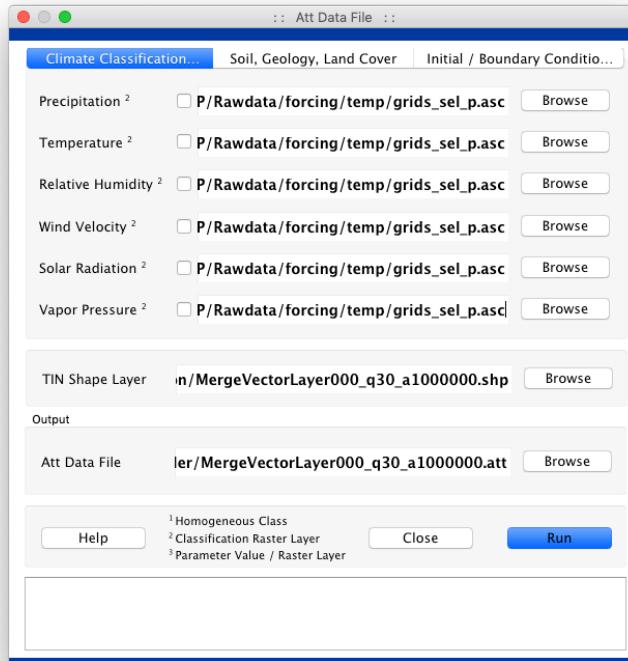
635

636 9.3 Att Data File

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

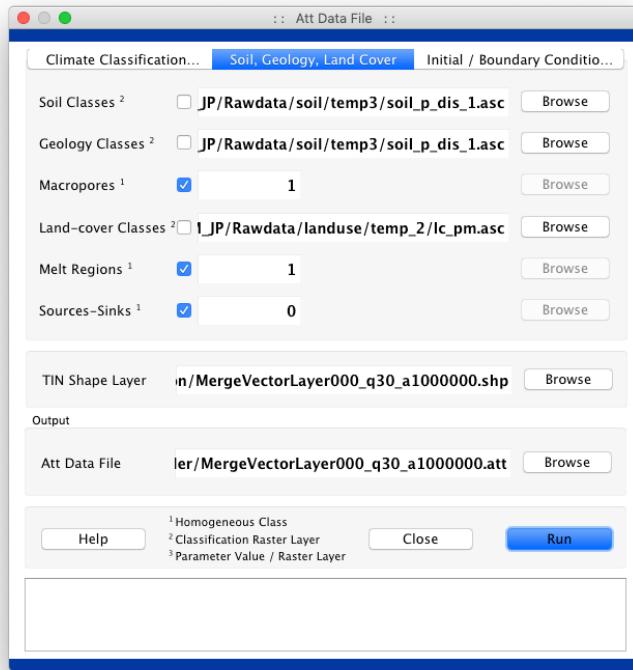
643 Raster layers saved in '.asc' format in section 8 are used here for
 644 soil, geology, land cover, and forcing. In absence of geology raster, soil
 645 raster is used in its place. Fig. 9.2 and 9.3 show the dialog boxes.

646



647
648

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



649
650

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

651
652 Table 9.3 shows the file structure of '.att', and Table 9.4 shows
653 description of the variables listed in Table 9.3.
654

Table 9.2 File structure of ' .att'

Index	Soil	Geol	LC	IS_LC	Snow_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...								

655
656
657
658
659
660

661

662

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

663

664

9.4 Riv Data File

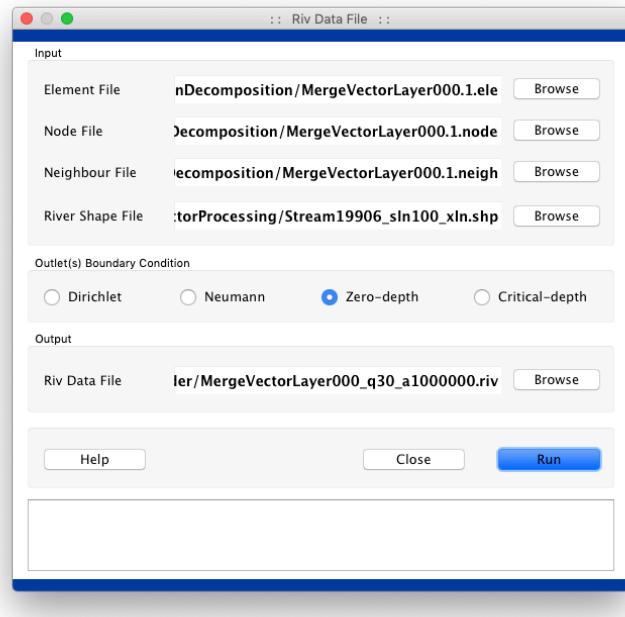
665 Topological information related to river segments is stored in this file.

666 Fig. 9.4 shows the dialog box for generating this file in PIHMgis, and

667 Table 9.5 and 9.6 shows the structure and descriptions of variables in

668 the generated '.riv' file.

669



670
671

Figure 9.4 Riv Data File dialog

672

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										

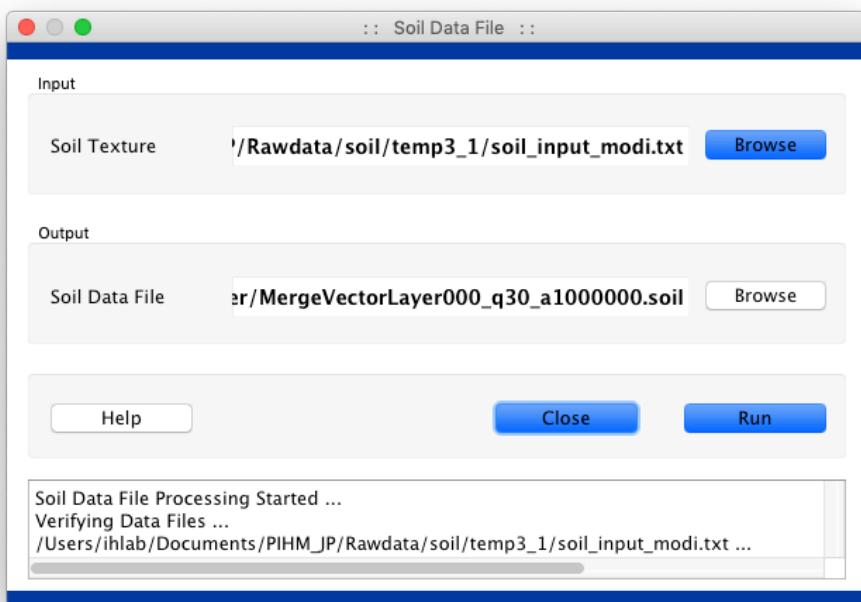
674

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:
 $[D=ax(W/2)^b]$

677 **9.5 Soil Data File**
678 Soil module uses pedo-transfer functions (Wösten, J.H.M. et al., 2001)
679 to estimate soil hydraulic parameters from soil texture data. Soil
680 hydraulic parameters required by the PIHM are: vertical soil hydraulic
681 conductivity, porosity, residual porosity, macropore depth, macropore
682 vertical hydraulic conductivity, van-Genuchten drainage (alpha and
683 beta) parameters. For using pedo-transfer functions, following soil
684 textual properties are needed: (1) silt percentage, (2) clay percentage,
685 (3) organic matter (optional), and (4) bulk density (optional). Fig. 9.5
686 shows the dialog of soil module. Fig. 9.6 shows the soil input text file
687 for this module. This text file is based on the extracted soil properties
688 from SSURGO database that you've obtained in section 8.2. The file
689 should look like as shown in Fig. 9.6. Table 9.7 shows the format of the
690 input text file shown in Fig. 9.6. Table 9.8 shows the file structure of
691 '.soil', which is the output of this module and input to the PIHM model.
692 Table 9.9 shows description of variables in '.soil'.
693



694
695 *Figure 9.5 Soil Data File dialog*

696

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	28.90	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

697
698

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

699
700
701

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

702
703

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...							

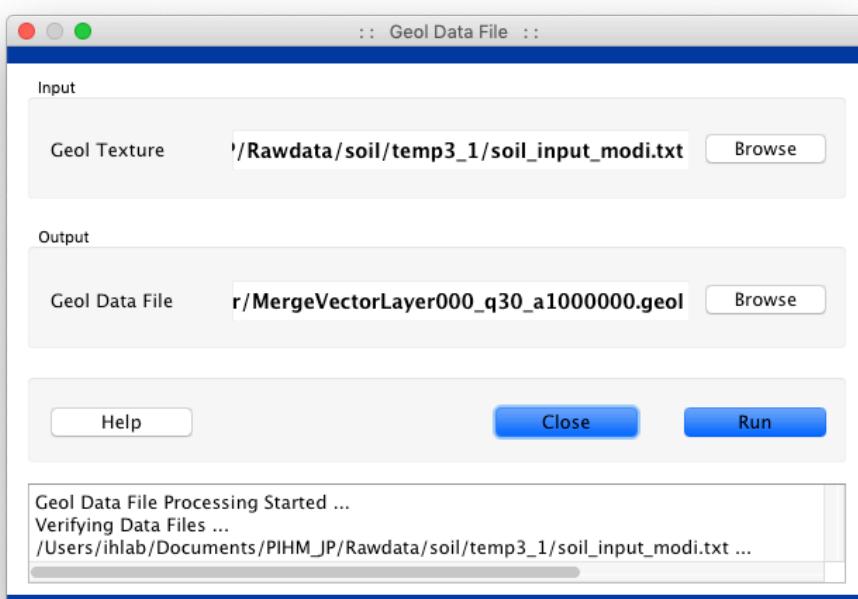
704
705

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 $K_u = S^{0.5} \left(1 - \left(1 - S^{\frac{\beta}{\beta-1}} \right)^{\frac{\beta-1}{\beta}} \right)^2$			

708 **9.6 Geol Data File**

709 If you have any form of measured data regarding hydraulic properties
710 of the geology, you can use 'Geol Data File' module to generate one.
711 This module is quite similar with 'Soil Data File' module you worked on
712 in section 9.5. It also uses pedo-transfer function (Wosten, J.H.M. et al.,
713 2001) for generating several parameters. You can use exactly same text
714 file that you used in 'Soil Data File', if you do not have geology
715 information. Fig. 9.6, as the input text file for 'Geol Data File' module.
716 Fig. 9.7 shows the dialog of this module. Table 9.10 shows the file
717 structure of '.geol', which is the result of this module. Table 9.11 shows
718 the description of variables in '.geol'.
719



720
721 *Figure 9.7 Geol Data File dialog*

722
723 *Table 9.10 File structure of '.geol'*

NumGeol									
Index	KsatH	KsatV	ThetaS	ThetaR	infD	Alpha	Beta	vAreaF	mackSatH
Index	KsatH	KsatV	ThetaS	ThetaR	infD	Alpha	Beta	vAreaF	mackSatH
...									
...	Repeat NumGeol times...								

724
725 *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	

726

727 **9.7 LC Data File**

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

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

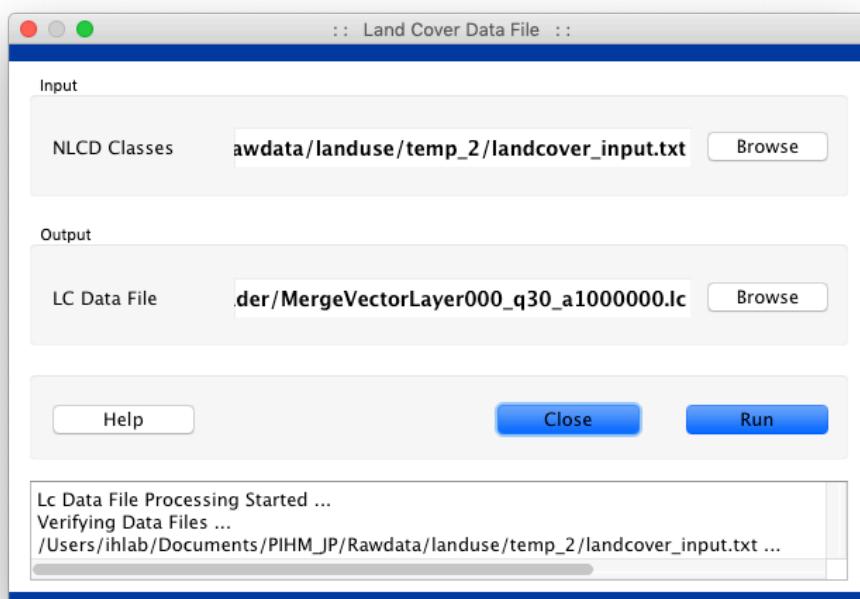
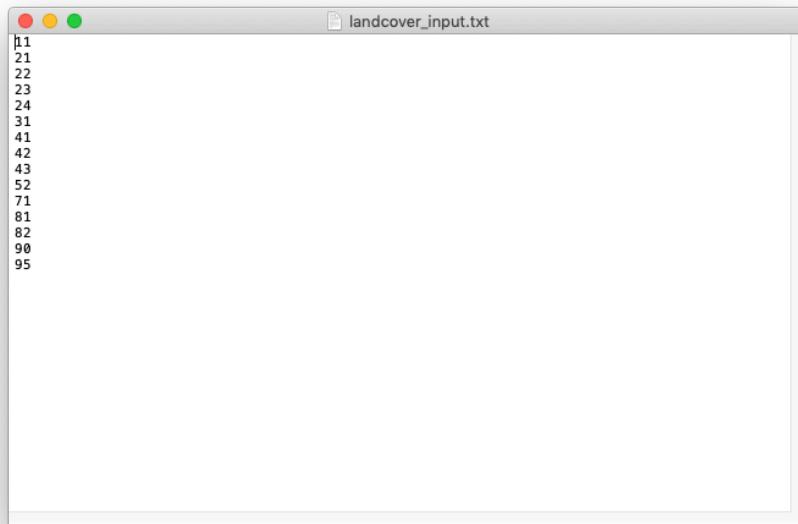
738
739

Figure 9.8 Land Cover Data File dialog

740



741

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

742

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

Class_1
...
...
Class_n

743

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
...							
...							
...							
Index	LAImax	Rmin	RS_ref	Albedo	VegFrac	n	RzD

747

Table 9.14 Description of variables in '.lc'

748

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}

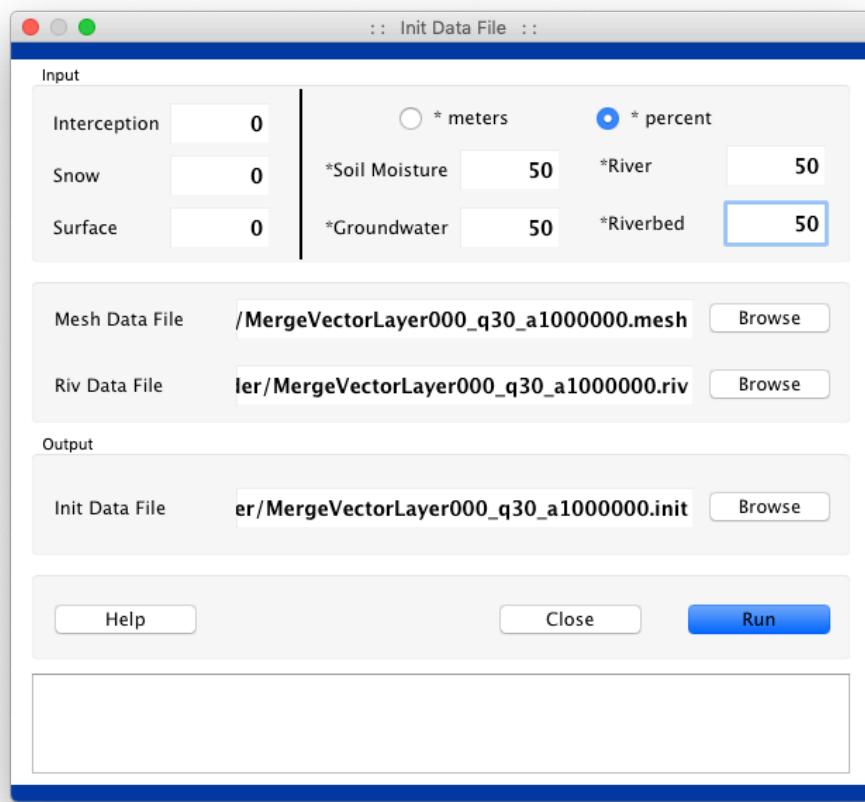
RzD	Double	Root zone depth	
-----	--------	-----------------	--

749

750 9.8 Init Data File

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

762



763
764

Figure 9.10 Init Data File dialog

765

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

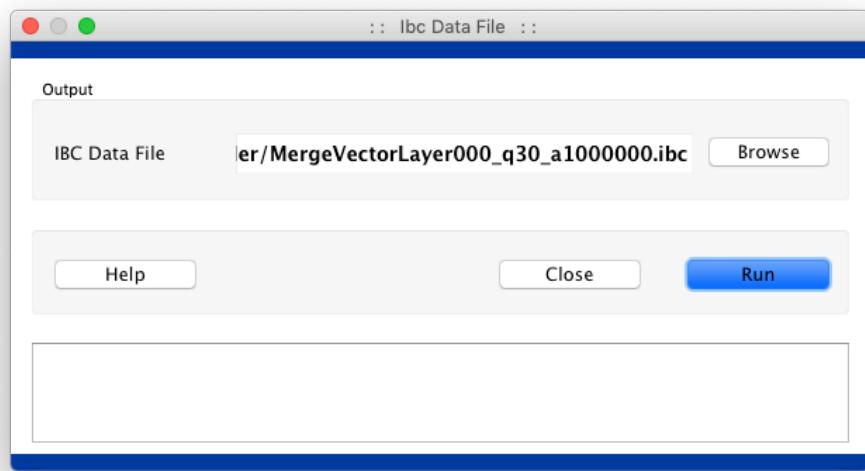
770 columns for each elements when it generated from PIHMgis v3.0.
771 However, now 6 columns are needed for the current version of
772 PIHM. As shown in the Table 9.15, 5 columns are for interception,
773 snow, overland, unsat, and sat. But PIHMgis v3.0 has 3 layer
774 (surface zone, unsaturated zone,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

778
779 *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.

780
781 **9.9 IBC Data File**
782 This module generates '.ibc' that contains any boundary condition
783 information if present for the modeling domain. PIHM has provision to
784 specify both Dirichlet and Neumann boundary conditions. The scope of
785 this module is limited to just creating a dummy file, which is generally
786 the case for a modeling domain. It generates a dummy file, which allows
787 setting up a simulation. If you have any information for this file, you can
788 modify the 'ibc' file based on Table 9.16 and 9.17.
789



790
791

Figure 9.11 Ibc Data File dialog

792

793

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

794
795*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)

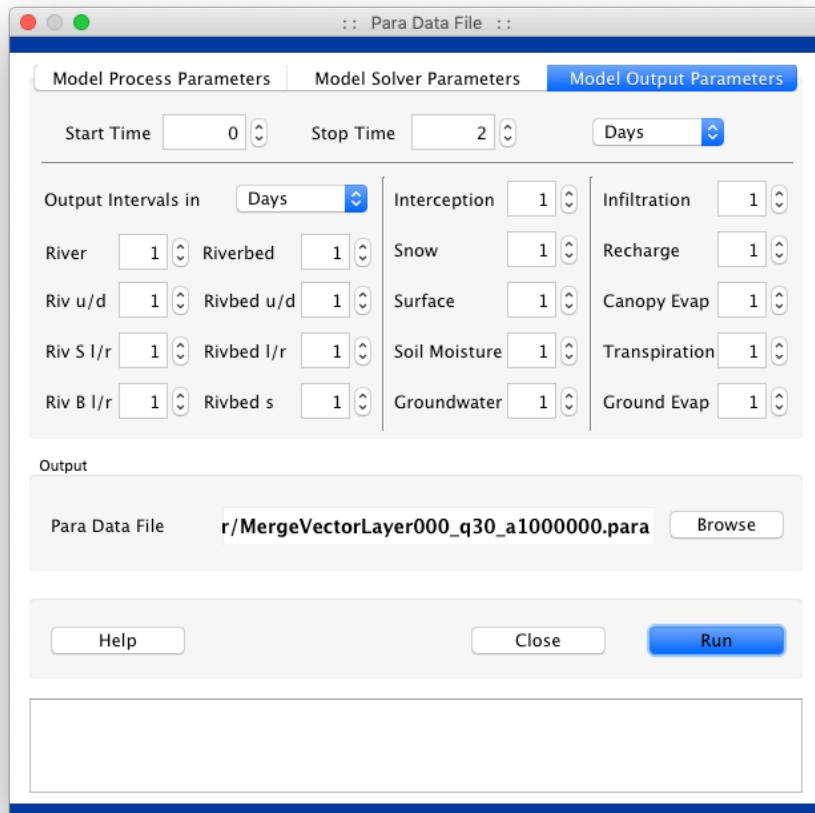
796

797 **9.10 Param Data File**

798 Parameter file provides all the control information for running PIHM. It
799 contains solver options, model modes, and parameters that govern
800 model error. You can see the several model and control panel from the
801 Table 9.18 and Table 9.19. With that, you can choose appropriate
802 models and parameters that are provided on the dialog as shown in Fig.
803 9.12. Fig. 9.12 only shows 'Model Output Parameters' tab, but you can
804 also set your configurations discussed above at 'Model Process
805 Parameters' tab and 'Model Solver Parameters' tab.

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

811



812
813

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[0]	AbsTol[1]	AbsTol[2]	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[i]	Double	Absolute Tolerances, i=0 for surface, i=1 for unsat and unsatT layer, i = 2 for groundwater	
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 x a ⁱ			

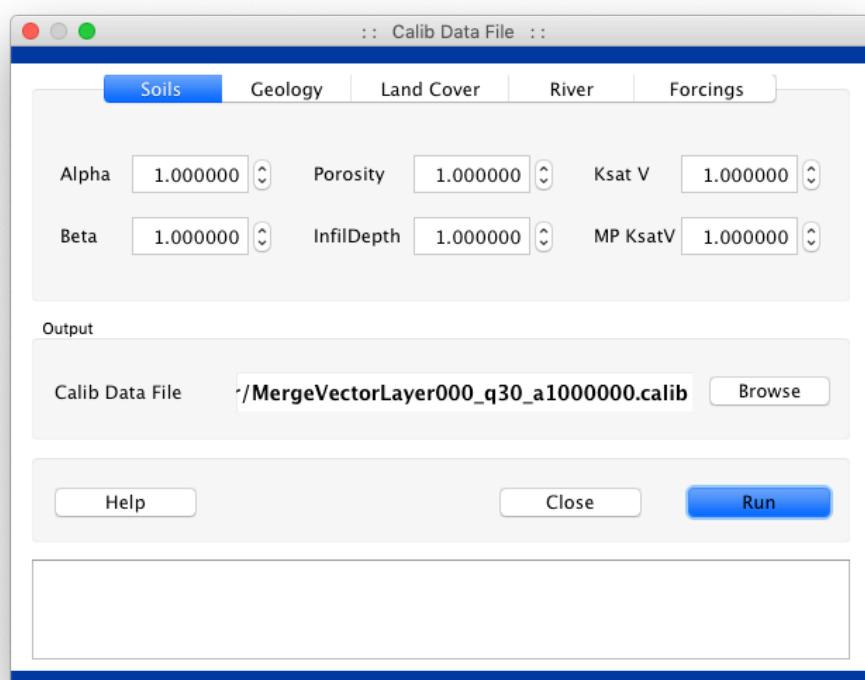
817

818

819 **9.11 Calib Data File**

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

825 On the dialog as shown in Fig. 9.13, you will notice a default value
826 has been provided for most of the parameters. A value 1.0 implies that
827 it would not change the parameter (since it is a multiplier to any
828 parameter). You can accept these default values for the purpose of first
829 run and come back later if you decide to change any of them.
830



831
832 *Figure 9.13 Calib Data File dialog*

833

834

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			

835

836

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	

837

838 9.12 Forc Data File

839 There is no module for generating forcing file in PIHMgis. This tutorial
 840 provides detail regarding how to generate forcing file with Table 9.22
 841 and 9.23 and captures of sample forcing file from Fig. 9.14 to 9.17.
 842 Sample forcing file uses NLDAS-2 data¹⁰ of one year (1.1.2017-

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

843 12.31.2017) for six grids (354,85-356,86), first discussed in section 3.2
844 and downloaded in section 8.4. Program codes used in this tutorial for
845 extracting data from NLDAS-2 grib files and generating '.forc' file from
846 such data are detailed in the Appendix A.

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

852 For "LAI" and "RL", the 'Index' follows the id number of land cover
853 types generated in section 8.3. These time series can be obtained from
854 remote sensing time-series layers by averaging values on each land
855 cover types repeatedly. Otherwise, you can use the table of NLDAS
856 vegetation parameters¹¹ as this tutorial did. It provides averaged LAI
857 values and RL values for each UMD LC class for each month. With this
858 table, you can write like Fig. 9.16. IsFactor was assumed as 0.0002 for
859 all land cover types.

860 Finally, this tutorial assumed there is no snow in the target area, so MF
861 is set to 0 as shown in Fig. 9.17. Furthermore, this tutorial also assumed
862 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 structure 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 1 st 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
```

867
868

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

869

```
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	

873
874

Figure 9.16 Screenshot of '.forc': LAI data

875

876

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		

877

Figure 9.17 Screenshot of '.forc': MF data

879 9.13 projectName File

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

883 10 Running PIHM

884 10.1 PIHM codes

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

898 10.2 Sundials

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

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

916

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

917
918

919 **10.3 Compiling PIHM using Makefile**

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

926

927
928

Figure 10.1 'makefile' for PIHM

929 After adjusting 'makefile' to your path of Sundials and running option,
930 you can compile PIHM with following commands.
931

932
933

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

934 Then, you would get executive file, 'pihm', in current folder. It is shown
935 in Fig. 10.2.
936

```
# Version: 2.0
# Date: "Nov, 2007"
#
# Programmer: Mukesh Kumar (muk139@psu.edu)@ PSU
#
# Makefile for PIHM
#
# ccode/pihm/Makefile.
#
# -----
#
# SHELL = /bin/sh

srkdir      =
builddir    =
top_builddir= ../../
top_builddir= ../../
prefix      = /home/jpark102/lib/sundials-2.4.0
exec_prefix = ${prefix}
includedir  = /home/jpark102/lib/sundials-2.4.0/include
libdir      = /home/jpark102/lib/sundials-2.4.0/lib

CPP          = /usr/bin/cc -E
CPPFLAGS     =
CC          = /usr/bin/gcc
CFLAGS       = -g -O0
#CFLAGS      =
LDFLAGS     =
LIBS        = -lm

# -DSUB_SURF_RIV is used while solving all the components of the model (surface, subsurface and river states)
# -DDIFFUSION is used for diffusion wave approximation of St. Venant's equation. Otherwise kinematic wave assumption will be used
# MACRO      = -DSUB_SURF_RIV
# MACRO      = -DSUB_SURF_RIV -DDIFFUSION -DLAYER3
# MACRO      = -DSUB_SURF_RIV -DDIFFUSION -DLAYER3 -DDCPL_VFLUX
# MACRO      = -DSUB_SURF_RIV -DDIFFUSION -DLAYER2
# MACRO      = -DSUB_SURF_RIV -DKINEMATIC -DLAYER2
# MACRO      = -DSURF_RIV -DDIFFUSION -DLAYER2 -DCONTAM -DSED
# -DSURF_RIV is used to run the model without sub-surface component. Generally used for calibration purposes
# MACRO      = -DSURF_RIV -DDIFFUSION -DLAYER2
# -DNO_UNSAT is used to run the model without unsaturated zone component. Overlandflow is set to zero 0 initially. DY for overland flow, and unsat zone is set to 0. DY for groundwater = ElePrep. Our goal is to use -DSUB_SURF_RIV -DDIFFUSION -DLAYER2. Long simulations to steady state can be performed. Note: DY for overland should not be set to zero if we want to track where lakes and rivers will be created. Exfiltration will have to be incorporated then
# MACRO      = -DSUB_SURF_RIV -DDIFFUSION -DLAYER2 -DNO_UNSAT
SRC         = pihm.c read_alloc.c initialize.c is_sm_et.c f_decouple.c f.c f_functions.c flux_cal.c update.c print.c
#SRC       = pihm.c read_alloc.c initialize.c is_sm_et.c f_decouple.c f.c f_functions.c update.c print.c

COMPILER_PREFIX =
LINKER_PREFIX   =

SUNDIALS_INC_DIR = ${includedir}
SUNDIALS_LIB_DIR = ${libdir}
SUNDIALS_LIBS    = -lsundials_cvode -lsundials_nvecserial

# EXEC_FILES = cvdx cvdxe cvbx cvkx cvkxb cvdemd cvdemk

all:
    @echo
    @echo '      make pihm      - make pihm          '
    @echo '      make clean     - remove all executable files'
    @echo

pihm:
    @echo '...Compiling PIHM ...'
    @${CC} ${CFLAGS} -I${SUNDIALS_INC_DIR} -L${SUNDIALS_LIB_DIR} -o ${builddir}/pihm ${SRC} ${SUNDIALS_LIBS} ${MACRO}

clean:
    @rm -f *.o
    @rm -f pihm
```

1,1 All

```

[[jpark102@uahpc test02]$ vi makefile
[[jpark102@uahpc test02]$ make clean
[[jpark102@uahpc test02]$ make pihm
...Compiling PIHM ...
[[jpark102@uahpc test02]$ ls
190110_f.c          flux_cal.c.bkp      Path.txt    test02.calib_noet  test02.mesh
diffs               initialize.c       pihm        test02.calib_org   test02.mesh.0
f.c                 is_sm_et.c        pihm.c     test02.FluxSurf   test02.para
f.c.bkp             makefile         pihm.h     test02.forc      test02.para_1yr10min
f.c.bkp1            newlai_test02.forc poros.txt  test02.geol     test02.para_1yr1min
f.c.ver1            nosub_test02.riv print.c    test02.ibc      test02.para_1yr30min
f.c.ver2            org_f.c          projectName.txt test02.init     test02.para_1yr60min
f.c.ver3_20190114   org_f_decouple.c read_alloc.c test02.init_new  test02.riv
f_decouple.c        org_f_functions.c result      test02.init_old  test02.soil
f_functions.c       orglai_test02.forc submit_job.sh test02.att     update.c
f_functions.h       org_test02.calib  test02.calib  test02.init_org var.c
flux_cal.c          orgwidth_test02.riv
[[jpark102@uahpc test02]$ ]

```

Figure 10.2 Compiled executive file, 'pihm'

937
938

10.4 Running PIHM

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

941
942

% ./pihm

943
944

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

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

951

```

ihlab — jpark102@uahpc:~/pihm/test02 — ssh jpark102@uahpc.ua.edu — 110x50
[[jpark102@uahpc test02]$ cat projectName.txt
test02
[[jpark102@uahpc test02]$ ls
190110_f.c          org_f.c           test02.calib_noet  test02.lc          test02.rivFlx2
diffs               org_f_decouple.c   test02.calib_org   test02.mesh       test02.rivFlx3
f.c                 org_f_functions.c  test02.cen_new    test02.mesh.0    test02.rivFlx4
f.c.bkp             orglai_test02.forc  test02.cen_old   test02.mesh.topo  test02.rivFlx5
f.c.bkp1            org_test02.calib   test02.et0      test02.NetP      test02.rivFlx6
f.c.ver1            orgwidth_test02.riv  test02.et1      test02.P        test02.rivFlx7
f.c.ver2            outputPath.txt     test02.et2      test02.para     test02.rivFlx8
f.c.ver3_20190114   pihm              test02.FluxSub   test02.para_1yr10min test02.rivFlx9
f._decouple.c       pihm.c            test02.FluxSurf  test02.para_1yr1min  test02.soil
f._functions.c      pihm.h            test02.forc     test02.para_1yr30min test02.stage
f._functions.h      poros.txt         test02.geol     test02.para_1yr60min test02.subriv
flux_cal.c          print.c           test02.GW       test02.para_June2017 test02.surf
flux_cal.c.bkp      projectName.txt   test02.ibc      test02.RechT     test02.unsat
initialize.c         read_alloc.c     test02.init     test02.riv       test02.unsatT
is_sm_et.c          result            test02.init_new  test02.rivFlx0   update.c
makefile            submit_job.sh   test02.init_old  test02.rivFlx1   var.c
newlai_test02.forc  test02.att      test02.init_org  test02.rivFlx10
nosub_test02.riv    test02.calib
[[jpark102@uahpc test02]$ ./pihm
... PIHM 2.0 is starting ...
hi
Start reading in input files ...
1) reading test02.riv ... done.
2) reading test02.mesh ... done.
3) reading test02.att ... done.
4) reading test02.soil ... done.
5) reading test02.geol ... done.
6) reading test02.lc ... done.
7) reading test02.forc ... done.
8) reading test02.ibc ... done.
9) reading test02.para ... done.
Start reading in calibration file...
10) reading test02.calib ... done.

```

952
953

Figure 10.3 Running PIHM with project name 'test02'

954 **10.5 Additional tips**
955 If you want to see what parameter values are applied, then you can
956 modify the last part of 'initialize.c' to print initialized values on the
957 terminal. It is hard and time consuming to obtain parameters and
958 properties of each element using GIS. Rather than that, you can modify
959 program code to print what values were inserted in the model run.
960 The initializing code, 'initialize.c' in the current version of PIHM,
961 includes commented sentences to print area, porosity, and many
962 parameters and properties of each element and river segment. You can
963 simply uncomment them and use to print these values. If you modify
964 codes, then you should clean the executive file ('make clean') and
965 compile again ('make pihm') before running PIHM.
966

- 967 **References**
- 968
- 969 Bhatt, G., Kumar, M., & Duffy, C. J. (2014). A tightly coupled
970 GIS and distributed hydrologic modeling
971 framework. *Environmental modelling & software*, 62, 70-84.
- 972 Kumar, M., Bhatt, G., & Duffy, C. J. (2009). An efficient domain
973 decomposition framework for accurate representation of geodata
974 in distributed hydrologic models. *International Journal of*
975 *Geographical Information Science*, 23(12), 1569-1596.
- 976 Kumar, M., Bhatt, G., & Duffy, C. J. (2010). An object-oriented
977 shared data model for GIS and distributed hydrologic
978 models. *International Journal of Geographical Information*
979 *Science*, 24(7), 1061-1079.
- 980 GLCF, n.d.: UMD Land Cover Classification. Retrieved from
981 <http://glcf.umd.edu/data/landcover/>
- 982 Hydrology Group of PSU, n.d.: PIHMgis website. Retrieved from
983 http://www.pihm.psu.edu/pihmgis_home.html
- 984 Hydrology Group of PSU, n.d.: PIHM wetsite. Retrieved from
985 http://www.pihm.psu.edu/pihm_downloads.html
- 986 Hydrology Group of PSU, 2007: Input file format manual of
987 PIHM v2.0. Retrieved from
988 http://www.pihm.psu.edu/Downloads/Doc/pihm2.0_input_file_fo_rmat.pdf
- 989 Hydrology Group of PSU, 2010: Tutorial of V-Catchment
990 Watershed for PIHMgis v2.3. Retrieved from
991 http://www.pihm.psu.edu/PIHMgis_v2.3_Tutorial_VCatchment.pdf
- 992 Hydrology Group of PSU, 2015: SourceForge website for PIHM.
993 Retrieved from <https://sourceforge.net/projects/pihmmodel/>
- 994 Kumar, M. (2009). Toward a hydrologic modeling system,
995 https://etda.libraries.psu.edu/files/final_submissions/4388
- 996 Lawrence Livermore National Laboratory, n.d.: Sundials
997 website. Retrieved from
998 <https://computation.llnl.gov/projects/sundials>
- 999 MRLC, n.d.: National Land Cover Database 2011 (NLCD2011)
1000 Legend. Retrieved from
1001 <https://www.mrlc.gov/data/legends/national-land-cover-database-2011-nlcd2011-legend>
- 1002 NASA GES DISC, 2018: README document for North American
1003 Land Data Assimilation System Phase 2 (NLDAS-2) Product.
1004 Retrieved from
1005 <https://hydro1.gesdisc.eosdis.nasa.gov/data/NLDAS/README.NLDAS2.pdf>

1010 NASA GES DISC, n.d.: Data Access. Retrieved from
1011 <https://disc.gsfc.nasa.gov/data-access>

1012 NASA GES DISC, n.d.: NLDAS_FORA0125_H: NLDAS Primary
1013 Forcing Data L4 Hourly 0.125 x 0.125 degree V002. Retrieved
1014 from
https://disc.gsfc.nasa.gov/datasets/NLDAS_FORA0125_H_V002/summary

1015 NASA LDAS, n.d.: NLDAS Unified Mask. Retrieved from
1016 https://ldas.gsfc.nasa.gov/nldas/asc/NLDASmask_UMDUnified.asc

1017 NASA LDAS, n.d.: NLDAS Vegetation Parameters. Retrieved from
1018 <https://ldas.gsfc.nasa.gov/nldas/NLDASmapveg.php>

1019 Qu, Y., 2005: An integrated hydrologic model for multi-process
1020 simulation using semi-discrete finite volume approach. Ph.D diss.,
1021 Pennsylvania State University.

1022 Qu, Y., Duffy, C.J., 2007: A semidiscrete finite volume
1023 formulation for multiprocess watershed simulation. Water
1024 Resources Research, 43, W08419, doi:10.1029/2006WR005752.

1025 UC Davis California Soil Resource Lab, n.d.: SoilWeb_nc097.
1026 Retrieved from
1027 https://casoilresource.lawr.ucdavis.edu/soil_web/ssurgo.php?action=list_mapunits&areasyymbol=nc097

1028 UC Davis California Soil Resource Lab, n.d.: SoilWeb_nc159.
1029 Retrieved from
1030 https://casoilresource.lawr.ucdavis.edu/soil_web/ssurgo.php?action=list_mapunits&areasyymbol=nc159

1031 USDA NRCS, n.d.: Soil Data Development Toolbox. Retrieved
1032 from
1033 https://www.nrcs.usda.gov/wps/portal/nrcs/detail/soils/survey/geo/?cid=nrcs142p2_053628

1034 USDA NRCS, n.d.: Soil Data Viewer. Retrieved from
1035 https://www.nrcs.usda.gov/wps/portal/nrcs/detailfull/soils/survey/geo/?cid=nrcs142p2_053620

1036 USDA NRCS, n.d.: Web Soil Survey. Retrieved from
1037 <https://websoilsurvey.sc.egov.usda.gov/App/WebSoilSurvey.aspx>

1038 USGS, n.d.: TNM Download (V1.0). Retrieved from
1039 <https://viewer.nationalmap.gov/basic/>

1040 Wang, D., Liu, Y., & Kumar, M. (2018). Using nested
1041 discretization for a detailed yet computationally efficient
1042 simulation of local hydrology in a distributed hydrologic
1043 model. *Scientific reports*, 8(1), 5785.

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053 Wösten, J.H.M., Pachepsky, Ya.A., Rawls, W.J., 2001:
1054 Pedotransfer functions: bridging the gap between available basic
1055 soil data and missing soil hydraulic characteristics. Journal of
1056 Hydrology, 251, 123-150, doi:10.1016/S0022-1694(01)00464-4.
1057
1058

1059 **Appendix A: Generating forcing file from NLDAS-2 grib**
1060 Additionally, in this version of tutorial manual, two program codes are
1061 introduced to extract data from NLDAS-2 grib files and organize data
1062 into the format of '.forc' file, respectively.

1063 First is a C++ code, 'extractForcing.cpp'. This code opens NLDAS-
1064 2 grib files and saves information of target grids and target duration into
1065 text files. This code consists 'main', 'stepOneHour', and 'MainFunction'.
1066 To set target grids and target duration, you should modify two things.
1067 First, you should modify lines shown in Fig. A.1. There should be the
1068 number of target grids, x-y grid index, and target duration. Second, you
1069 should find sentences defining characters of 'NLDAS_FOLDER' and
1070 'OUT_FOLDER', and modify them for your folder paths. Compiling this
1071 code ('gcc extractForcing.cpp -o a.out') and running it will result in text
1072 files shown in Fig. A.2.
1073



```
32 int main(){
33
34     int col[6];
35     int row[6];
36
37     row[0]=354;
38     col[0]=85;
39     row[1]=354;
40     col[1]=86;
41     row[2]=355;
42     col[2]=85;
43     row[3]=355;
44     col[3]=86;
45     row[4]=356;
46     col[4]=85;
47     row[5]=356;
48     col[5]=86;
49
50     mainFunction( 2015, 1, 1, 0, 2017, 12, 31, 23, 6, row, col);
51
52 }
```

1074
1075 *Figure A.1 Target grids and duration in 'extractForcing.c'*

1076



```
[ualjxp@dmcvlogin1:ext_yan> ls
ncl_to_forc  x354y85zPP.txt  x355y85zRH.txt  x355y85zRN.txt  x356y85zTT.txt  x356y85zVP.txt
x354y85zPP.txt  x354y85zRH.txt  x355y85zRN.txt  x355y85zTT.txt  x356y85zVP.txt  x356y85zWD.txt
x354y85zRH.txt  x354y85zRN.txt  x355y85zTT.txt  x355y85zVP.txt  x356y85zWD.txt
x354y85zRN.txt  x354y85zTT.txt  x355y85zVP.txt  x355y85zWD.txt  x356y85zPP.txt
x354y85zTT.txt  x354y85zVP.txt  x355y85zWD.txt  x356y85zPP.txt  x356y85zRH.txt
x354y85zVP.txt  x354y85zWD.txt  x355y85zPP.txt  x356y85zRH.txt  x356y85zRN.txt
x354y85zWD.txt  x355y85zPP.txt  x355y85zRH.txt  x356y85zRN.txt  x356y85zTT.txt
ualjxp@dmcvlogin1:ext_yan>
```

1077
1078 *Figure A.2. Result files using 'extractForcing.c'*

1079
1080 Second code is a ncl code, 'make_forc_1.ncl'. It reads extracted
1081 text files and additional files for monthly LAI and roughness length

1082 values, and generates 'sample.txt' that includes all forcing and
 1083 vegetation time-series information within '.forc' format. To
 1084 generate the result file, you should modify two things. First, as
 1085 shown in Fig. A.3, there are grid numbers, length of duration, and
 1086 paths. Grid number should be same with previous code, and
 1087 length of duration should be hours. Paths should be modified in
 1088 whole code.
 1089

```

begin
; load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
;;;;
setting
grids = new((/6,2/),"integer")
grids(0,:) = (/354, 85/)
grids(1,:) = (/355, 85/)
grids(2,:) = (/356, 85/)
grids(3,:) = (/354, 86/)
grids(4,:) = (/355, 86/)
grids(5,:) = (/356, 86/)
vars = (/"PP", "TT", "RH", "WD", "RN", "VP/")
;var_names = (/"Prep", "Temp", "RH", "Wind", "Rn", "Vp/")
;lnth = 17544
lnth = 8760*3

;;;;
file open, write first line
fname = "/home/ualjxp/NLDAS/ext_yan/ncl_to_forc/sample.txt"
system("rm -f " + fname)
header1 = "6 6 6 6 0 6 14 1 0"
;header2 = "Time PP"
hlist = [header1]
write_table(fname, "w", hlist, "%s")
;write_table(fname, "a", [header2], "%s")

;;;;
do i_var=0,2
do i_grid=0,5
  z1 = asciread("~/NLDAS/ext_yan/x" + grids(i_grid,0) + "y" + grids(i_grid,1) + "z" + vars(i_var)
+ ".txt", (/lnth,4/), "float")

```

1090
 1091

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

1092 Second, as shown in Fig. A.4, you should modify settings for
 1093 vegetation parameters, LAI and RL. Additional files, 'rl_nldas.txt'
 1094 and 'lai_nldas.txt' includes LAI and RL values for 12-month
 1095 columns and 14-land-cover rows. Check the paths of these
 1096 vegetation parameter files are right. You can also modify
 1097 'tstep_veg', the number of records, and 'jday', time for each
 1098 record. Setting in Fig. A.4 is for 3-year forcing file. You can change
 1099 in any values, but you should check first 'jday' value is 0 and the
 1100 last 'jday' value is the last day's Julian date.
 1101

```

;;;;
tstep_veg = 2*12*3
jday = (0, 30, 31, 58, 59, 89, 90, 119, 120, 150, 151, 180, 181, 211, 212, 242, 243, 272, 273, 30
3, 304, 333, 334, 364, 365, 395, 396, 423, 424, 454, 455, 484, 485, 515, 516, 545, 546, 576, 577, 607
, 608, 637, 638, 668, 669, 698, 699, 730, 731, 761, 762, 789, 790, 820, 821, 850, 851, 881, 882, 911,
912, 942, 943, 973, 974, 1003, 1004, 1034, 1035, 1064, 1065, 1095)
rl_db_nldas = asciread("~/NLDAS/ext_yan/ncl_to_forc/rl_nldas.txt", (/14,12/), "float")
lai_db_nldas = asciread("~/NLDAS/ext_yan/ncl_to_forc/lai_nldas.txt", (/14,12/), "float")

```

1102
 1103

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').

1119

```

|| Made by Junhyeong Park, Dec 19 2018
// 80% of code was copied from 'analysiscalculation.c' that Dr. Kumar Mukesh gave it to me
// Org. code was for calculating representative values of whole domain based on results
// I added precipitation and simple water budget calculation

#include <stdio.h>
///#include <stdlib.h>
///#include <math.h>
///#include <string.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 Interval 60 // Time interval length in minute recorded in result files, which is indicated in .pa ra file
#define NumRiv 128 // Num. of river segments
#define RivOutlet 1 // River id of the outlet river segment
main()
{
    double sumArea=0.0,val,totVar0=0,totVar1=0,totVar2=0,totVar3=0,totVar4=0,totVar5=0,totVar6=0,aveg eT0=0,avegt1=0,avegt2=0,totet0=0,toteti1=0,toteti2=0,totetsum=0,avgprec=0,totprec=0,wb=0,area[NumEle],rivare a[NumRiv],val0[NumTS][NumRiv],val1,val2,val3,val4,val5,val6,et0,et1,et2,prec,poros[NumEle][3],init[NumEle][6],porosRiv[NumRiv],initriv[NumRiv][2],tvar;
    int i,j;
    FILE *fpin0,*fpin1,*fpin2,*fpin3,*fpin4,*fpin5,*fpin6,*fparea,*fprivarea,*fpresult,*fpporos,*fpin it,*fpet0,*fpet1,*fpet2,*fpprec;^M

    // reading model result files
    fpin0=fopen("test02.rivFlx1","r");
    fpin1=fopen("test02.unsat","r");
    fpin2=fopen("test02.unsatT","r");
    fpin3=fopen("test02.GW","r");
    fpin4=fopen("test02.stage","r");
    fpin5=fopen("test02.sabriv","r");
    fpin6=fopen("test02.surf","r");
    fpet0=fopen("test02.et0","r");
    fpet1=fopen("test02.et1","r");
    fpet2=fopen("test02.et2","r");
    fpprec=fopen("test02.P","r");

    // reading model input file
    fpinit=fopen("test02.init","r");

    // reading files including initial information of the model (can be obtained by modifying 'initialization.c')
    fparea=fopen("/home/jpark102/pihm/test02/result/ancillarypihmcodespostprocessing/back/area.txt","r");
    fprivarea=fopen("/home/jpark102/pihm/test02/result/ancillarypihmcodespostprocessing/back/rivarea.txt","r");
    fpporos=fopen("/home/jpark102/pihm/test02/result/ancillarypihmcodespostprocessing/back/poros.txt","r");

    // opening the balance analysis result file
    fpresul=t=fopen("test02.wb","w");^M

    for(i=0;i<NumEle;i++)
    {

```

1120
1121

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

1122 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.

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