



TERM PROJECT FINAL REPORT

IMPLEMENT GIS AS A TOOL TO PREPARE
INPUTS DATA OF WRF_HYDRO TO RUN
AND COMPARE WITH NWM OUTPUTS
FOR LOGAN RIVER WATERSHED

BY: IRENE GAROUSI-NEJAD

PHD. STUDENT, CIVIL AND ENVIRONMENTAL ENGINEERING, UTAH STATE UNIVERSITY
CEE 6440, FALL 2016

INTRODUCTION

Streamflow prediction has been an imperative issue and has remained an unsolved scientific problem among hydrologists. Nowadays, with a changing climate and land cover, people experience more prolonged droughts and floods over the United States. This fact highlights the need for more accurate hydrologic models to predict streamflow. Among efforts, the most recent endeavors to predict streamflow has led to developed, launched, and unveiled the America's first National Water Model (NWM) on August 16. The underlying hydrologic model for NWM is the Weather Research and Forecast model for Hydrology (WRF-Hydro). WRF-Hydro has been developed to facilitate the representation of various spatially distributed variables/parameters and terrestrial processes with the option of choosing different parameterization. As regards, it can potentially be used as a hydrologic model to study streamflow prediction.

The key objective of this study, as a term project for CEE 6440, was to start working with WRF_Hydro, a physically-based hydrologic model, and evaluating the streamflow forecast (simulation) in Logan River watershed during the month of August 2016. Moreover, this project aimed to compare the results of WRF-Hydro with NWM outputs (as the best estimates of current streamflow conditions). WRF-Hydro is a sophisticated hydrologic model which requires various inputs. In this project, the ArcGIS platform is implemented to generate some of the required inputs of WRF-Hydro model.

STUDY AREA

The Logan River watershed is a small watershed within the hydrographic¹ Great Basin as shown in Figure 1. This figure includes the geographic dataset representing the national geospatial framework, NFIE-Geo, provided by Fagan (2015) and can be found on HydroShare web server as a public and sharable geodatabase file. According to the attribute table of the Great Basin shape file, the area of Great Basin is approximately 367,000 square kilometers that drains internally. All precipitation in this the region evaporates, sinks underground, or flows into lakes (such as the Great Salt Lake). The Logan River Watershed, shown in Figure 1, was delineated using ArcGIS software in Exercise 4 of CEE6440 and is used in this project.

The 30-meter resolution National Elevation dataset (NED30m) of the Logan River watershed, downloaded from the server URL <http://elevation.arcgis.com/arcgis> by using the ArcGIS server user connection, is shown in Figure 2. According to the Figure 2, it can be seen that the watershed is bounded by the Bear River Range to the west, the northwestern, and a small part to the east. The highest elevation is about 3000 meters at the mountains and the lowest elevation is about 1400 meters high to which all the streamflow within this watershed drains. According to Figure 2, it can be seen that the Logan River starts from the north and passes through the center of watershed towards the southwestern (following the dark blue

¹ The word hydrographic emphasize that we are looking at the way the water flows. Because the Great Basin can be defined hydrographically, topographically – the way the landscape formed, or biologically – the way the resident plants and animals exists (<https://www.nps.gov/grba/planyourvisit/the-great-basin.htm>).

color). The outlet of the Logan River water is shown with a black circle in Figure 2. This point is the location of one of the USGS streamflow gages, called Logan River above State Dam with the code identifier as 10109000. It is important to mention that the “North_America_Albers_Equal_Area_Conic” projection system is used (recommended in hydrology working) to display and analyze this project.

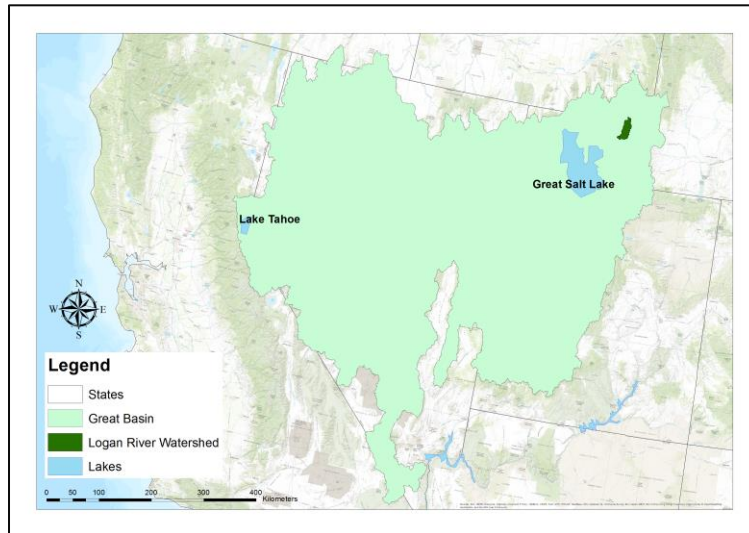


Figure 1. Map of Great Basin and Logan River Watershed

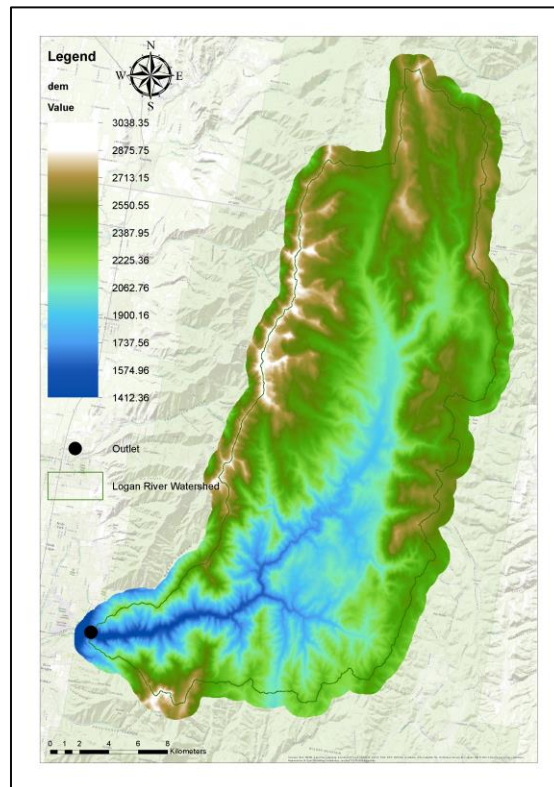


Figure 2. Topography map and the outlet of Logan River Watershed

MODELS AND TOOLS

This section addresses the models and tools that were used during this project. At the sub-section A, a description of WRF_Hydro model workflow, inputs, and outputs are presented. Then, second sub-section B introduces the GEOGRID_STANDALONE ArcGIS toolbox to produce some other required inputs of the WRF_HYDRO model.

A. WRF_HYDRO MODEL

WRF-Hydro has been developed by National Center for Atmospheric Research (NCAR) to facilitate the representation of various spatially distributed variables/parameters and terrestrial processes with the option of choosing different parameterizations in order to simulate and predict terrestrial water processes across a wide range of spatial and temporal scales (https://www.ral.ucar.edu/projects/wrf_hydro). The WRF-Hydro modeling system provides a means to couple hydrological model components to atmospheric models and other Earth modelling architecture (Gochis et al., 2015). WRF-Hydro has been applied for a wide range of research and operational prediction problems both in the U.S. and abroad (Yucel et al., 2015; Senatore et al., 2015).

Figure 3 illustrates the modeling system of WRF-Hydro. According to Figure 3, it can be seen that the main box of WRF-Hydro is connected to two dashed boxes, namely module and mode. The former includes six main modules and the latter shows three different modes. The main physical processes are performed within the modules which are (1) Colum Land Surface Model, (2) Subsurface Flow Routing, (3) Overland Flow Routing, (4) Channel Routing, (5) Lake and Reservoir Routing, and (6) Base-flow Routing. The dashed box namely “mode” emphasizes the statement as mentioned previously that WRF-Hydro can be used as a coupled or uncoupled mode. Users have the options to activated/deactivated each of the modules and choose the mode that they require for their case studies. In this project, I activated the first four modules and chose the uncoupled mode as shown in grey boxes in Figure 3 for the simplification of the case study. In Figure 3, the dotted yellow box represents the Land Surface Model (LSM) and the dotted green box shows all the modules related to the Routing Terrain model (RT). The reason for separating the LSM and RT is related to one of the aspects of WRF-Hydro that uses different set of grid cell size (scales) for LSM and RT such that the LSM can be run on coarser grid cells size and RT can be run on finer grid cells size. Although WRF-Hydro routing functions are typically run on a finer grid than the land model, the two different grids must be compatible. The factor that relates these two grid cell sizes are called aggregation or regridding factor (AGGFACTR) and is shown in Figure 4.

A brief description of the WRF-Hydro modules, used in this project, are provided in Table 1 and more detailed about the physics description of WRF-Hydro modules can be found on Gochis et al. (2015).

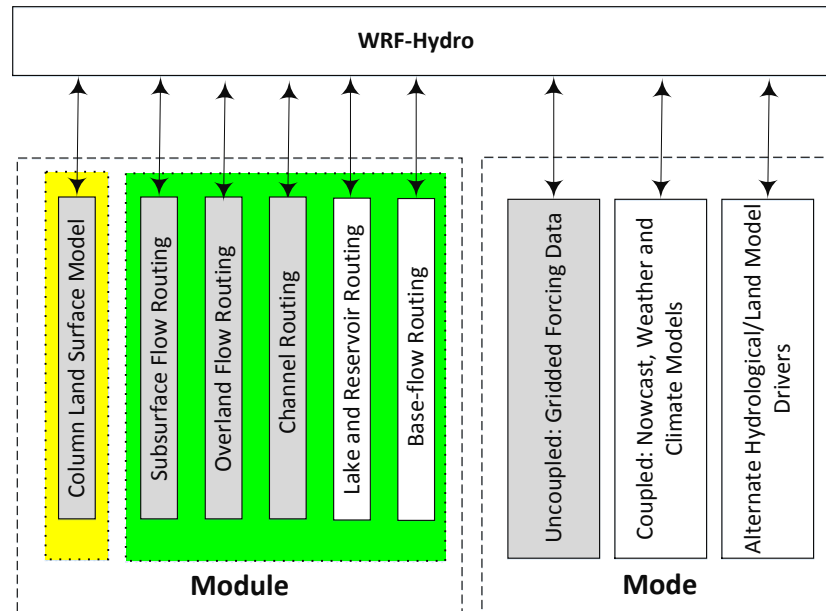


Figure 3. Sketch of the WRF-Hydro modular calling structure

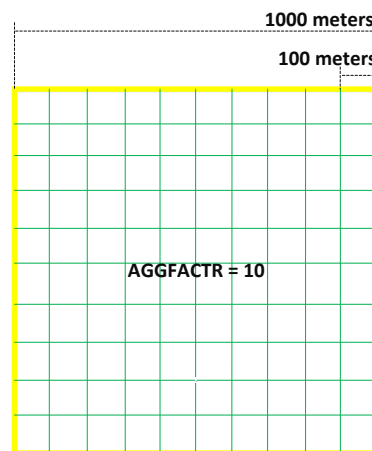


Figure 4. Nesting of RT grid cells (green boxes) within a LSM grid cell (yellow box).

The required inputs data for WRF-Hydro model operation can be categorized into (1) domain processing and static input datasets, (2) time-evolving (“forcing”), (3) parameters. Each one of these inputs are described as follows.

- **Domain processing and static input datasets**

The simulation domain (LSM) and its required static inputs are built using the GEOGRID program which is one of the WRF Pre-processing System (WPS) programs. In fact, the use of GEOGRID program is extremely useful since it automates the entire procedure of defining simulation domain, geo-referencing and attributing most of the LSM parameter data required to execute the Noah LSM in WRF-Hydro. GEOGRID acquires and interpolates land surface terrain, soils and vegetation data from standard, readily available data products (such as the USGS

National Elevation Dataset or the STATSGO Soils Database) to the LSM domain. To run GEOGRID, it is required to define some parameters so that the GEOGRID can be able to create a proper simulation domain. In this project, the size of each grid cell in LSM is defined 1000 meters. The number of cells is 59(rows)*59(columns) and the projection is set to be Lambert conformal. It is important to mention that WRF model currently support four types of projections (Lambert conformal, Mercator, Polar stereographic, and Cylindrical equidistance) among which Lambert is used since it is recommended for the mid latitudes (http://www2.mmm.ucar.edu/wrf/users/tutorial/201607/wps_general.pdf). As we have learned in this course (CEE6440, 2016) the true latitude 1 and true latitude 2 are required to be defined as shown in Figure 5. In this project the true latitude 1 and true latitude 2 were 60 (i.e. 60°N) and 20 (i.e. 20°N), receptively.

Table 1. Description of WRF-Hydro modules

Module	Description
Land Surface Model	The newest LSM used in WRF-Hydro is NOAHMP Land surface model which is a state of the art, community, 1-dimensional land surface model that simulates soil moisture (both liquid and frozen), soil temperature, skin temperature, snowpack depth, snowpack water equivalent, canopy water content and the energy flux and water flux terms at the earth's surface (Mitchell et al., 2005).
Subsurface flow	In WRF-Hydro, subsurface lateral flow is calculated prior to the overland flow to allow exfiltration from fully saturated grid cells to be added to the infiltration excess calculated from the LSM. This will update the value of surface head prior to routing of overland flow.
Overland flow	In WRF-Hydro, the overland flow is calculated when the depth of water on a model grid cell exceeds a specified retention depth.
Channel	IN WRF-Hydro, the overland flow reaching grid cells identified as "channel" grid cells pass a portion of the surface water in excess of the local ponded water retention depth to the channel model.

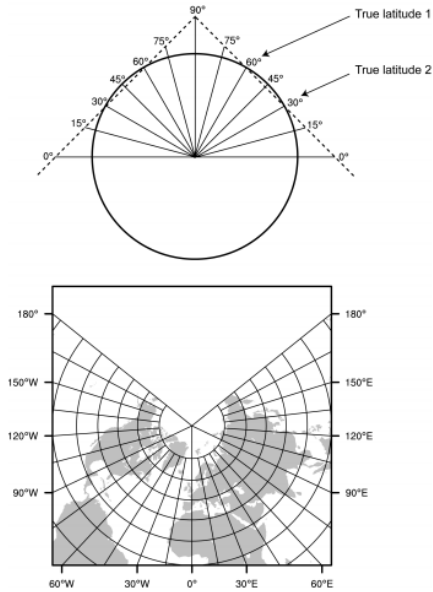


Figure 5. Lambert conformal projection system

After the LSM domain is created, the GEOGRID program starts to acquire and interpolate data required by LSM. These data are Topographic elevation in units of metes, latitude, longitude, land use fraction in units of fraction, top layer soil texture category in units of fraction, monthly mean green vegetation fraction values in units of fraction, and monthly mean surface albedo values in units of % (not including snow effect). Overall, all of these interpolated data have the same resolution as LSM domain and are within a NetCDF file. Using “Make NetCDF Raster Layer” tool in ArcGIS, each variable in the NetCDF file can be seen. For example, the topographic elevation is shown in Figure 6.

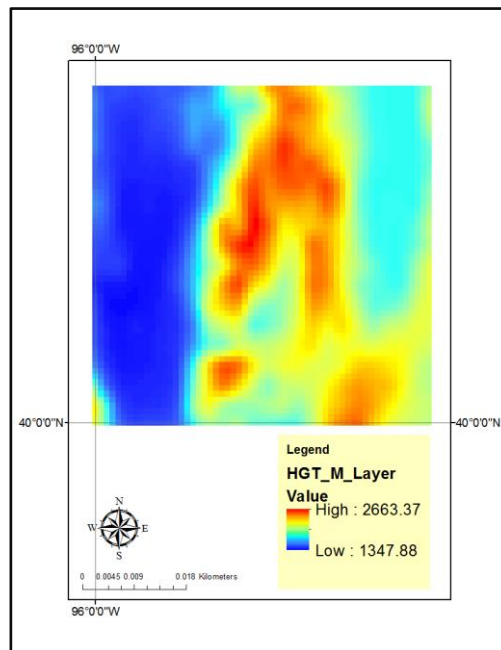


Figure 6. Interpolated topographic elevation data as one of the result of GEOGRID program

It is worth mentioning that since these variables are within a NetCDF file, they do not have projection system individually. Therefore, if we look at the source of the topographic elevation, it is unknown. Therefore, that is why the Figure 6 is not titled (as it is expected with the Lambert conformal projection) due to the projection.

- **Time-evolving (meteorological forcing data)**

Modern land surface hydrology models, such as WRF-Hydro, require meteorological forcing data to simulate land-atmosphere exchanges as well as terrestrial hydrologic processes. The common input forcing data for the NOAH LSM are incoming shortwave radiation, incoming longwave radiation, specific humidity, air temperature, surface pressure, near surface wind (horizontal components), and liquid water precipitation rate. In uncoupled mode, the meteorological forcing data must be provided as gridded input time series. Among different forcing data input options in WRF-Hydro, two options, (1) Idealized and (2) the hourly North America Land DATA Assimilation Systems (HRLDAS) input file, were used separately in this project.

The Idealized option is the simplest method to force the model and requires no input files. In this option, a simple rainfall event is prescribed in the model of 25.4 millimeters (one inches) per hour for one-hour duration. The rest of the forcing data variables are set to have either constant values or in the case of temperature and radiation variables, a fixed diurnal cycle. This option is mainly used for simple testing of the model and is convenient for checking whether or not components besides the forcing data are properly being read into the model and working (Gochis et al., 2015).

In HRLDAS option, all meteorological variables are packed into one NetCDF file for each time step (i.e. 1 hour). These data were acquired using the shell scripts provided by NCAR and can be found on (https://www.ral.ucar.edu/projects/wrf_hydro) in LINUX platform.

- **Parameters**

To run the WRF-Hydro, some text files which include parameter values are required to be defined (based on the LSM model which is used). In this project, since the NOAHMP is used as the LSM, the parameters are as follows.

CHANPARAM.TBL: including initial depth of water in the channel, channel side slope, Manning's roughness coefficient, and channel bottom width for 10 different stream orders.

GENPARAM.TBL: including NOAH LSM general parameter table.

HYDRO.TBL: hydraulic parameters used in WRF-Hydro.

SOILPARAM.TBL: contacting soil physical parameters indexed by soil textural classes.

MPTABLE.TBL: containing vegetation parameters indexed by land use/land cover categories.

These files are the default values and have not been changed for this project.

In addition to specifying the LSM domain with the use of GEOGRID, it is required to define the RT domain (high resolution terrain grid). The RT grid specifies the data that are necessary to route water across the landscape (via overland and saturated subsurface flow) and through stream channels. Generating the RT grid is thoroughly described in the next sub-section using the ArcGIS.

B. GEOGRID_STANDALONE TOOL

Hydrologic application of Geographical Information System (GIS) have ranged from synthesis and characterization of hydrologic techniques to prediction of response to hydrologic events (Bruce and Arlen, 1993) have been used extremely in hydrology many researchers in hydrology (Tarboton, 1997; Prasad et al., 2006). Nowadays, with the use of Python scripting language, Python toolboxes can be implemented in ArcGIS as a way to create custom geo-processing tools directly from Python scripts, a great integrator and fairly easy to work with in spatial programming. The GEOGRID_STANDALONE tool is a Python toolbox developed by Sampson and Gochis (2015) and can be used by within ArcGIS 10.1 and newer versions. It is important to mention that using this toolbox requires to activate the ArcGIS Spatial Analyst Extension.

The objective of the GEOGRID_STANDALONE tool is to create the geospatial and tabular data layers (some as NetCDF and some as text formats) as inputs for terrestrial overland flow, subsurface flow, and channel routing processes (as shown in Figure 7) required by WRF_Hydro model. This tool can be download under the download tab (at the section of General Pre-Processing Utility Scripts) from URL server https://www.ral.ucar.edu/projects/wrf_hydro and can be used readily through ArcCatalog under the path at which it was downloaded. As shown in Figure 7, the GEOGRID_STANALONE.pyt¹ contains two toolsets, namely Processing and Utilities. The former contains a single script tool, namely ProcessGeogridFile and the latter contains multiple script tools which are used to perfume additional functions to prepare or review of outputs from the ProcessGeogridFile script tool. In this project I only used ProcessGeogridFile script.

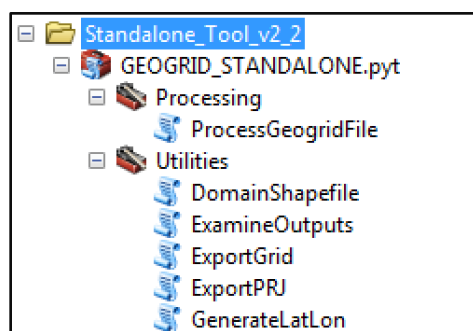


Figure 7. Tree view of the GEOGRID_STANDALONE toolbox in ArcCatalog

¹ .pyt is the file name extension for a Python toolbox.

The ProcessGeogridFile requires several inputs (mandatory and optional) and some default parameters shown in Figure 8. The first input is the GEOGRID NetCDF file which was generated using GEOGRID program (as mentioned previously this file includes the simulation domain of LSM along with required soil and vegetation data). The second input (which is optional) is a comma-delimited file including the locations of points at which the user would like to see the results of WRF-Hydro as hydrographs. If this input is not defined, the results will be only provided at the outlet of the domain. In this project, I only intended to see the result at the outlet of the Logan River watershed. Therefore, no input is defined here. The next input is the DEM raster file (30-meter resolution). The next input is related to the value of the regrading factor (as mentioned previously this number defines how the RT grid cells nest on LSM grid cells). In this project, this value was considered as 10. Hence, the RT grid cell size was 100 meters (coming from 1000 meters divided by 10). The last input is related to the number of pixels which is used to define stream. This threshold is defined as 200.

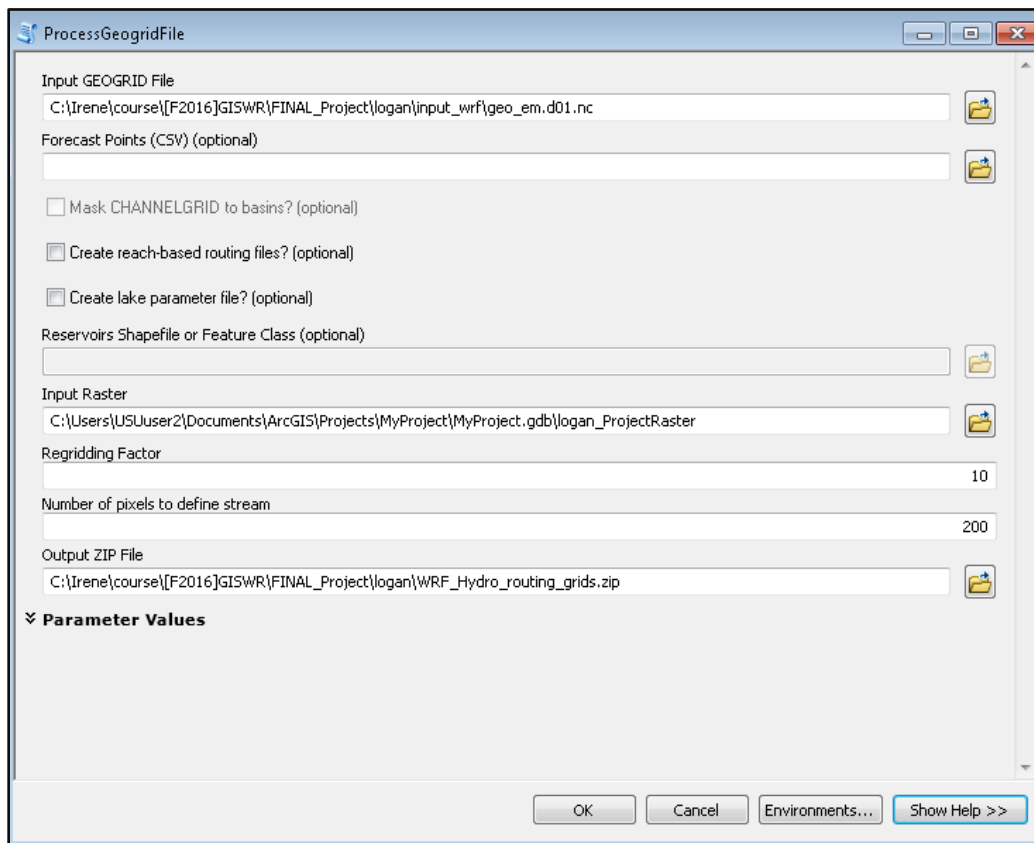


Figure 8. The ProcessGeogridFile tool and its required inputs

Eventually, after running, the output file (which is a zip file) is generated with the default name of WRF_Hydro_routing_grids.zip. All of the NetCDF files in this zip file are then used as inputs in running WRF-Hydro. The important point is that these NetCDF files requires to be concatenated into a single NetCDF file using the shell script (ncks_file_concatenator.sh) which can be found on https://www.ral.ucar.edu/projects/wrf_hydro. This final NetCDF file includes

all the inputs that are required for the RT modules. This NetCDF file can be viewed in ArcGIS using “Make NetCDF Raster Layer” tool. Figure 9 and 10 depict the topography data and channel grid data for high-resolution terrain routing calculation in WRF-Hydro. The number of grid cells of these files is 10 (regridding factor) times bigger than LSM grid cell size. Therefore, there is 590*590 cells with the resolution of 100 meters.

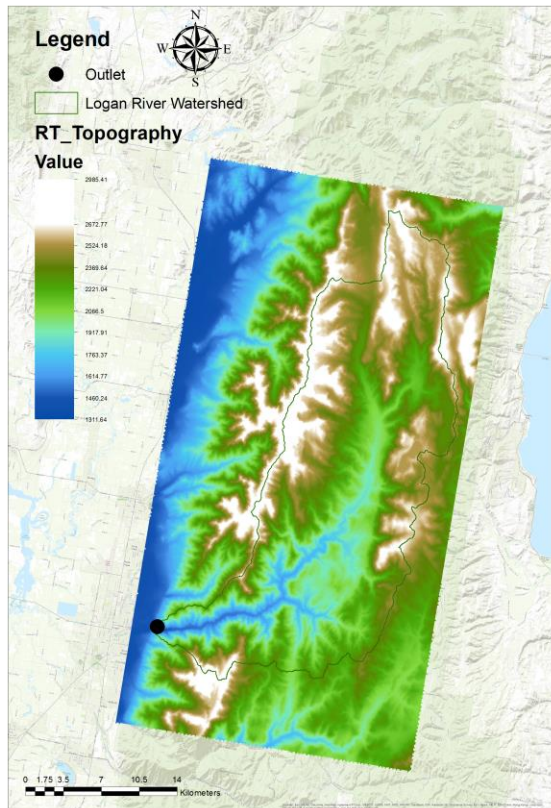


Figure 9. High-resolution topography data of Logan River Watershed used in WRF-Hydro

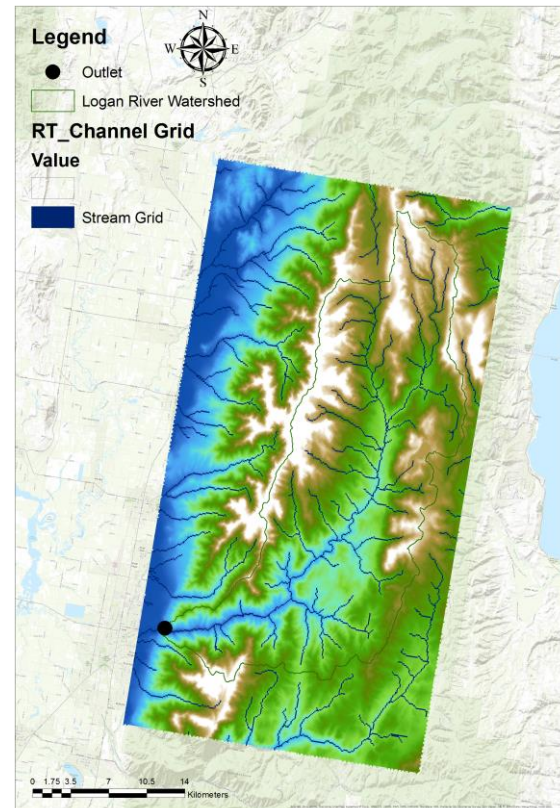


Figure 10. Channel grid data of Logan River Watershed used in WRF-Hydro

Though these files are retrieved from a NetCDF, the interesting issue is that they have a proper projection (Lambert).

METHODOLOGY

Previous sections describe the main structure of WRF-Hydro and all the processes which were required to provide the WRF-Hydro inputs. After these steps, all the inputs were put in proper paths and defined in “namelist.hrlas” and “hydro.namelist” text files and WRF-Hydro was run using cluster of the Center of High Performance Computing (CHPH) of the University of Utah. The results are discussed in the next section.

RESULTS AND DISCUSSION

Running the WRF-Hydro was a challenging process since various methods should have been used to provide the inputs data and even with the inputs there were many errors related to running WRF-Hydro on LINUX platform. I spent most of my time during this semester to learn how to define the inputs and how to deal with the errors while running WRF-Hydro. Though there were big progress upon solving the errors, the model was successfully run for the idealized scenario at which a precipitation event of 25.4 millimeters per hour happened for a duration of one hour. This indicate that all the model components besides the forcing data are properly being read into the model and working, which was a good news for me.

There is a R package developed by [McCreight, 2016](#) which can be used to visualized the results of WRF-Hydro. Figure 11 is depicted using rwrhydro package. More details about this package can be found at (<https://github.com/mccreigh/rwrhydro/blob/master/README.Rmd>). A part of this code which was used in this project is shown in Appendix.

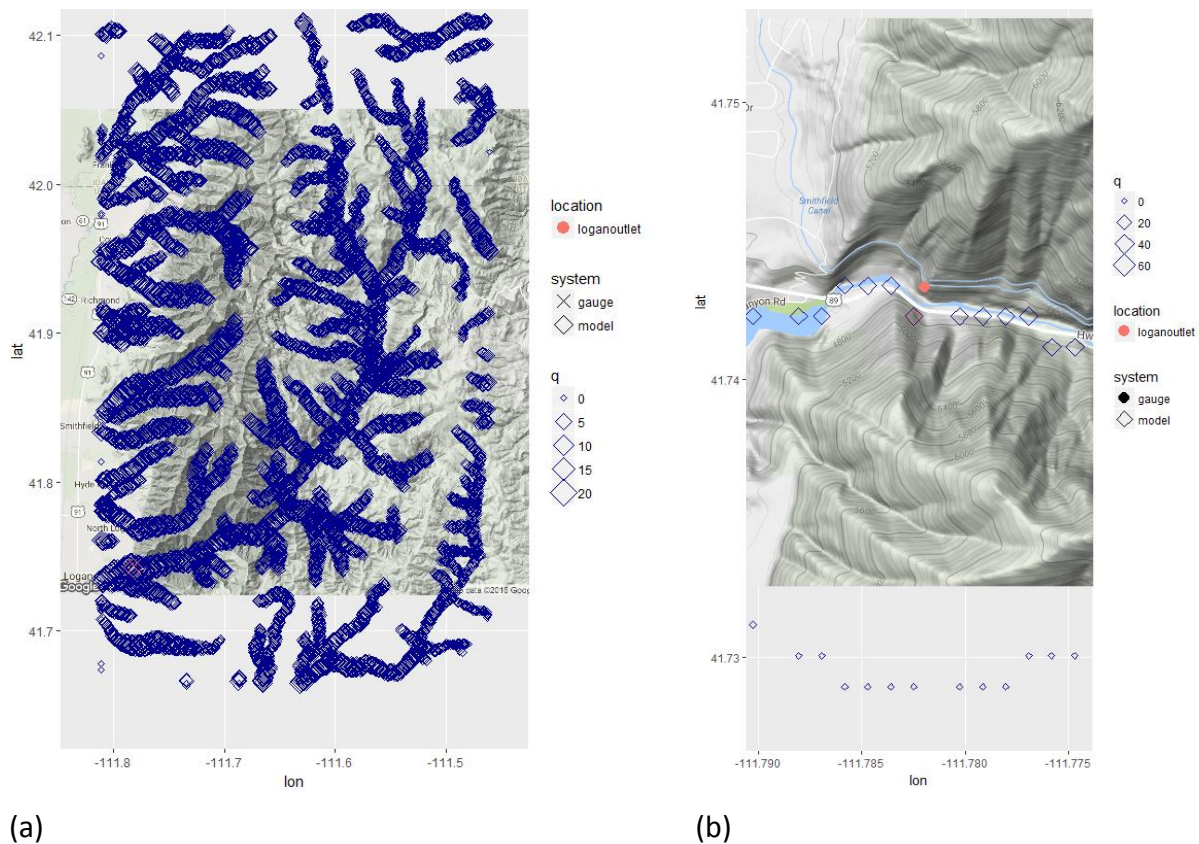
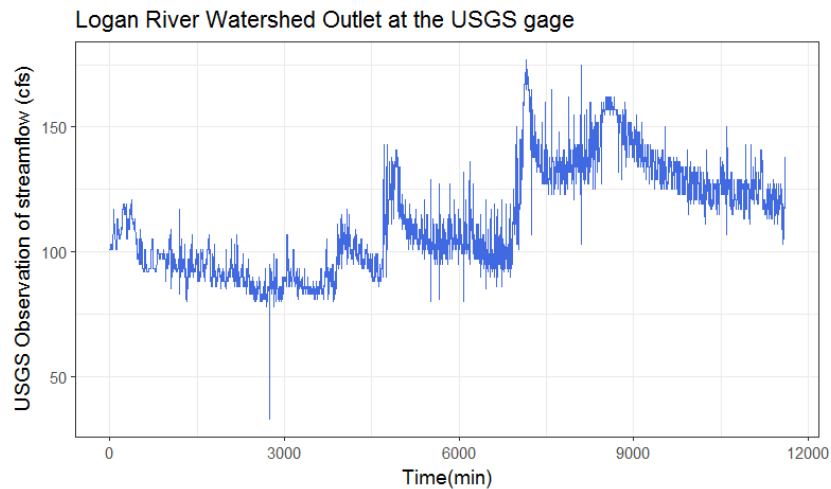
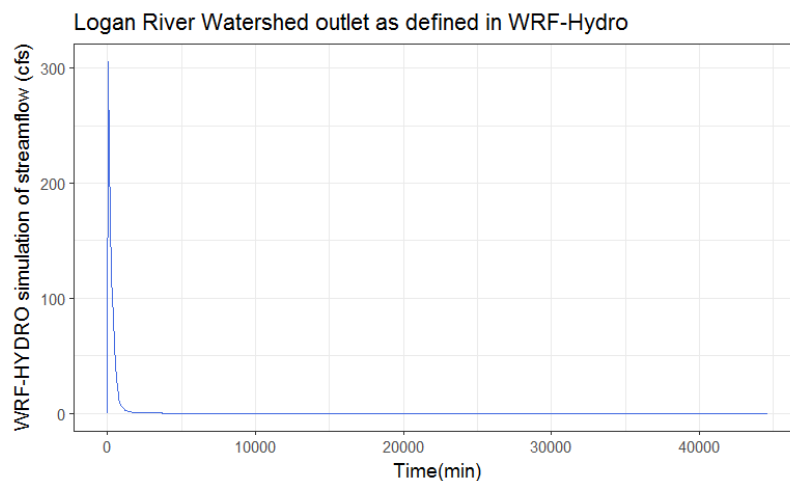


Figure 11. The visualization of channel network grids and the outlet of the Logan River Watershed using rwrhydro for (a) the whole domain and (b) zoomed in to the outlet location. [q is zero at the grids near the outlet]

As it can be seen in Figure 11(b), the location of the USGS gage, Logan River above State Dam (shown as a red point) differs from the grid cell (shown as a red square) which is defined in the model. The results of streamflow for these two locations (one representing the observation values and the other representing the simulated or forecasts values) are shown in Figure 12.



(a)



(b)

Figure 12. (a) observed and (b) simulated streamflow (considering the idealized scenario for forcing data) values for the Logan River watershed starting from the August 1 and ending at August 31 of the year 2016

The idealized scenario is revealed in the Figure 12(b). Between time 0 to 60 minutes (the first hour of simulation period), a rainfall event with the intensity of one inches occurred at the first hour of simulation on the first day (August 1) and its effect is seen as the peak value of

about 300 cubic feet per second at the second hour of simulation on the first day. Then, the hydrograph decreased such that after 5 hours it reached a third of its peak (~100 cubic feet per second) and then continued to decrease. At the middle hours of the second day it reached to the values below 1 cubic foot per second and at the end of the last time steps its value became near 0.001 cubic feet per second.

Previously in the study area section, it was mentioned that the drainage area of the Logan River is about 378,000 square kilometers. The total amount of the streamflow during the simulation time (one month) is about 1349.376 cubic feet per second or 38.175 cubic meters per second. I did some simple calculation to see whether the values of the streamflow are generated correctly.

$$\text{Precipitation Rate} = PR = 0.0254m / hr$$

$$\text{Precipitation Duration} = 1hr$$

$$\text{Flow} = 38.175m^3 / hr$$

$$\text{Area} = 367,000km^2$$

$$\text{Runoff depth} = \frac{\text{Flow}}{\text{Area}} = \frac{38.175 \times 3600 \times 24 \times 31}{367000 \times 1000000} = 0.017m$$

$$P = 0.0254m > R = 0.017m$$

The second scenario, using HRLDAS forcing data, could not be run correctly and there were errors. I spent too much time understanding why the generated forcing data cannot be used by WRF-Hydro. One reason might be the process of generating HRLDAS is not performed correctly, though I generate them without any errors. Another possible answer might be that the newest version of WRF-Hydro (specifically the "namelist.hrlDas") cannot recognize the HRLDAS. I did contact many peoples during this semester to overcome this issue and I would like to appreciate their guidance, however, this issue still remains and I will continue working on this issue.

CONCLUSION

As a term project, this work intended to implement ArcGIS as a tool to prepare some inputs of WRF-Hydro model. As regards, the stand-alone Python pre-processing utility that is designed to assist users in the creation of WRF-Hydro routing grids was used. Using this toolbox was beneficial in terms of automating the creation functions (specifically related to the routing grid) that WRF-Hydro requires. Other inputs, related to the LSM and forcing data were generated using shell scripts. Two scenarios for the type of forcing data were considered. The idealized scenario, which is the simplest option, was successfully run and the results show that all the components of the model work fine. The second scenario which uses the HRLDAS data could not be run and it is still remaining a challenge. This project helped me start working with WRF-Hydro and learning more about its physics. Throughout this project, I learned how to produces different inputs of WRF-Hydro using several tools, such as Geogrid Standalone tool.

REFERENCE

- Bruce, A. D. and Arlen, D. F. (1993). "Review of GIS application in hydrologic modeling", *Journal of Water Resources Planning and Management*, 119(2).
- Fagan, C. (2015). NFIE-Geo Great Basin Region, HydroShare, <http://www.hydroshare.org/resource/46c1b0f6c16c4ae1b71298cda7cfc0ab>
- Gochis, D. J., W. Yu, D. N. Yates. (2015). "The WRF-Hydro model technical description and user's guide, version 3.0. NCAR Technical Document", 120 pages. Available online at: http://www.ral.ucar.edu/projects/wrf_hydro/
- Mitchell, K. (2005). "The community Noah land surface model (LSM)", User's Guide, http://www.ral.ucar.edu/research/land/technology/lsm/noah/Noah_LSM_USERGUIDE_2.7.1.pdf.
- Prasad, A., D. G. Tarboton, C. H. Luce and T. A. Black, (2006), "A GIS Tool to Analyze Forest Road Sediment Production and Stream Impacts," AGU Hydrology Days, Colorado State University, Fort Collins, March 20-22.
- Sampson, K. and Gochis, D. J. (2015). "WRF-Hydro GIS Pre-Processing Tools, Version 2.2". Available online at: https://www.ral.ucar.edu/projects/wrf_hydro
- Tarboton, D. G., (1997), "A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models," *Water Resources Research*, 33(2): 309-319.
- Yucel, L., Oena, A., Yilmazb, K. K., and Gochis, D. J. (2015). "Calibration and evaluation of a flood forecasting system: Utility of numerical weather prediction model, data assimilation, and satellite-based rainfall," *Journal of Hydrology*, 523, 49-66.
- Senatore, A., Mendicino, G., Gochis, D. J., Yu, Wei, Yates, D. N., and Kunstmann, H. (2015), "Fully coupled atmosphere-hydrology simulations for the central Mediterranean: impact of enhanced hydrological parameterization for short and long time scales", *Journal of Advances in Modeling Earth System*, 7(4), 1693-1715.

APPENDIX

```

library(devtools)
install.packages("readr") # this install dependency curl as well.
devtools::install_github("mccreigh/rwrhydro", ref='devBranch')
# download ncdf for windows-64 : http://cirrus.ucsd.edu/~pierce/ncdf/
# use the following and in the window choose the zip file downloaded.
# install.packages(file.choose(), repos=NULL, type = "binary")

library(rwrhydro)
checkForUpdates()

installed.packages("ggplot2")
install.packages("colorspace")
library(ggplot2)
fcPath <- 'C:/Irene/research/rwrhydro/ByJames'
geoFile <- paste0(fcPath, '/logan_wthBcio/DOMAIN/geo_em.d01.nc')

# make sure packages rgdal and sp are installed.
# download rgdal from: https://cran.r-project.org/web/packages/rgdal/index.html
# use the following and in the window choose the zip file downloaded.
# install.packages(file.choose(), repos=NULL, type = "binary")
# download RgoogleMaps from: https://cran.r-project.org/web/packages/RgoogleMaps/index.html
# use the following and in the window choose the zip file downloaded.
# install.packages(file.choose(), repos=NULL, type = "binary")
# download png package from: https://cran.r-project.org/web/packages/png/index.html
# use the following and in the window choose the zip file downloaded.
# install.packages(file.choose(), repos=NULL, type = "binary")
# download geosphere package from: https://cran.r-project.org/web/packages/geosphere/index.html
# use the following and in the window choose the zip file downloaded.
# install.packages(file.choose(), repos=NULL, type = "binary")
# download jpeg package from: https://cran.r-project.org/web/packages/jpeg/index.html
# use the following and in the window choose the zip file downloaded.
# install.packages(file.choose(), repos=NULL, type = "binary")
# proto, rjson
# install.packages(file.choose(), repos=NULL, type = "binary")
# install reshape2 from tools/install packages

install.packages("proto")
devtools::install_github("dkahle/ggmap")
library(ggmap)

# channel network
chrtFile <- paste0(fcPath, '/logan_wthBcio/201608010100.CHRTOUT_DOMAIN1')
LoClinkFun <- VisualizeChanNtwk(chrtFile)
args(LoClinkFun)
LoClinkFun(click=TRUE)
LoClinkFun <- VisualizeChanNtwk(chrtFile, exclude=350)
LoClinkFun(click=TRUE)
gaugePts <- list(loganoutlet = data.frame(lon=-111.7819444, lat=-41.7433333))
LoClinkFun <- VisualizeChanNtwk(chrtFile, gaugePts=gaugePts, exc=350, plot=FALSE)
LoClinkFun <- VisualizeChanNtwk(chrtFile, gaugePts=gaugePts, exc=350, plot=FALSE, gaugeAccuracy=17)
LoClinkFun()
LoClinkFun(pad=5, gaugeShape=16)
LoClinkFun(zoom=15, gaugeShape=16, gaugeZoom='loganoutlet', pad=15)

# USGS =====
dbPath <- 'C:/Irene/research/rwrhydro/ByJames/logan_wthBcio/usgs'
stndf <- FindUsgsStns(stnLon=-111.7819444, stnLat=-41.7433333, within=.001)
str(stndf)
huc8 <- GetSiteHuc(stndf$site_no)
str(FindUsgsStns(huc=huc8))
coData <- GetUsgsHucData(huc=huc8)
str(coData)
coFiles <- SaveHucData(coData, outPath=dbPath)
coFiles
QuerySiteInfo(c('station_nm', 'site_no', 'dec_lat_va', 'dec_long_va'), path=dbPath)
dataloganoutlet <- QuerySiteData(QuerySiteName("LOGAN RIVER ABOVE STATE DAM, NEAR LOGAN, UT", path=dbPath),
product='00060', path=dbPath)

str(dataloganoutlet)
timehour = seq(0, 11608)
library(ggplot2)
ts <- ggplot(data=dataloganoutlet, aes(timehour, dataloganoutlet$value)) +
  xlab("Time(min)") +
  ylab("USGS observation of streamflow (cfs)") +
  ggtitle("Logan River Watershed outlet") +
  geom_line(color="royalblue") +
  theme_bw(base_size=14)
ts

# WRF-HYDRO =====
dataPath <- 'C:/Irene/research/rwrhydro/ByJames/logan_wthBcio/'
modStrd.chrt.lg <- ReadFrxtPts(paste0(dataPath, 'frxst_pts_out2016n.txt'))
str(modStrd.chrt.lg)
library(plyr)
mod <- data.frame(modStrd.chrt.lg$timest, modStrd.chrt.lg$q_cfs)
str(mod)
timehour = seq(0, 44664, 60)
str(timehour)
library(ggplot2)
ts <- ggplot(data=mod, aes(timehour, modStrd.chrt.lg.q_cfs)) +
  xlab("Time(min)") +
  ylab("WRF-HYDRO simulation of streamflow (cfs)") +
  ggtitle("Logan") +
  geom_line(color="royalblue") +
  theme_bw(base_size=14)
ts

```