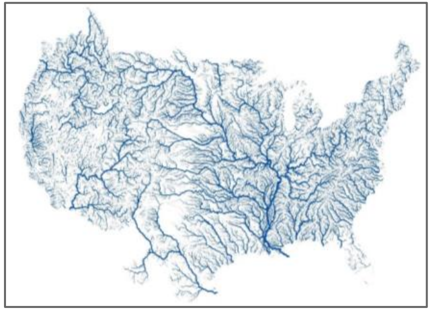
WRF-Hydro GIS Pre-processing Tool Overview



Kevin Sampson & Matt Casali

National Center for Atmospheric Research

Outline

- WRF-Hydro GIS Pre-processing tools
- Basic GIS terrain pre-processing for WRF-Hydro
- Demonstration: Generating WRF-Hydro Routing Grids

WRF-Hydro GIS Tools

- Pre-processing tools, written in Python
- 2 versions:
  - ArcGIS Python Toolbox for ArcGIS Desktop & ArcGIS Pro, using Spatial Analyst
  - Fully Open-source, using Whitebox Tools python API (WBT)
- Variety of WRF-Hydro configuration options supported
- Fast, efficient method for producing the ‘routing stack’ necessary to run WRF-Hydro
- Consistent processing methodology between domains, regions, datasets
- Provides WRF-Hydro with a complete set of hydrologically processed routing grids and spatial metadata
- Removes the heavy GIS burden from modelers

https://ral.ucar.edu/projects/wrf_hydro/pre-processing-tools

GitHub https://github.com/NCAR/wrf_hydro_gis_preprocessor

Whitebox Tools

GitHub https://github.com/jblindsay/whitebox-tools

- Command-line executable
- Open-source license (MIT)
    - Free to use, redistribute
- Parallel, cross-platform, no external dependencies
- Extensible using a Python API (WBT)
- Contains 445+ geo-processing tools, including many hydrological analysis and stream network analysis tools.
- Full documentation with literature citations.

python™ WhiteboxTools™ NumPy GDAL netCDF

Requirements (not strict)

- Conda (miniconda3, Anaconda)
  - Python 3.6
  - GDAL
  - NumPy
  - netCDF4
  - Pyproj
  - Whitebox 1.2.0

Create the environment in 1 line using ‘conda create’:

> conda create -n wrfh_gis -c conda-forge python=3.6 gdal netCDF4 numpy pyproj whitebox=1.2.0

How the Tools Work

- Python scripts wrapped to act as command-line executable

tool script
(e.g. Build_Routing_Stack.py)

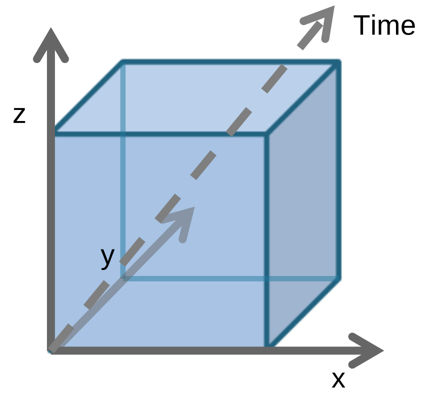
function script
(wrfhydro_functions.py)

Advantages:
- Easy to modify
- Portable

- Each script tool performs a particular pre-processing or utility function
- Parameter handling and validation
- Functionality can also be imported into custom python scripts

NetCDF File Format

- network Common Data Form
  - ".nc" extension
- Self describing
  - Includes information about the data coordinate system
- Machine independent
  - Usable in many operating systems
- Used extensively in Atmospheric Science
- Multidimensional
  - x,y,z,t



“Fulldom” File (Routing Grids)

- netCDF input file for WRF-Hydro
- “Full resolution” domain file (Fulldom_hires.nc)
- Stores all routing grids as 2-dimensional variables
- Stores CF-compliant spatial metadata
  - grid_mapping
  - Projection information
  - Coordinate System variable
  - GIS-compliant projection information
- Easy to import into GIS Applications (ArcGIS, QGIS)
- Ingested directly by WRF-Hydro

A Note on CF Metadata

- Climate and Forecast Conventions for netCDF data
  - Like a standard
  - Current version 1.8
  - http://cfconventions.org/latest.html
- CF conventions for just about any type of data
  - Gridded
  - Point
  - Profile
  - timeSeries
- CF-compliant netCDF files make them much easier to use in client applications
  - Panoply, ArcGIS, QGIS

Build Routing Stack

```
(wrfh_gis2) C:\Users\ksampson\Documents\GitHub\wrf_hydro_gis_preprocessor>python Build_Routing_Stack.py
Script initiated at Wed Oct 28 17:02:06 2020
usage: Build_Routing_Stack.py [-h] -i IN_GEOGRID [--CSV IN_CSV]
                              [-b BASIN_MASK] [-r RB_ROUTING]
                              [-l IN_RESERVOIRS] -d INDEM [-R CELLSIZE]
                              [-t THRESHOLD] [-o OUT_ZIP_FILE]
                              [-O OVROUGHRTFAC_VAL] [-T RETDEPRTFAC_VAL]
                              [--starts CHANNEL_STARTS] [--gw GW_POLYS]

This is a program to perform the full routing-stack GIS pre-processingfor WRF-
Hydro. The inputs will be related to the domain, the desired routing nest
factor, and other options and parameter values. The output will be a routing
stack zip file with WRF-Hydro domain and parameter files.

optional arguments:
  -h, --help            show this help message and exit
  -i IN_GEOGRID         Path to WPS geogrid (geo_em.d0*.nc) file [REQUIRED]
  --CSV IN_CSV          Path to input forecast point CSV file [OPTIONAL]
  -b BASIN_MASK         Mask CHANNELGRID variable to forecast basins?
                        [True/False]. default=False
  -r RB_ROUTING         Create reach-based routing (RouteLink) files?
                        [True/False]. default=False
  -l IN_RESERVOIRS      Path to reservoirs shapefile or feature class
                        [OPTIONAL]. If -l is TRUE, this is required.
  -d INDEM              Path to input high-resolution elevation raster
                        [REQUIRED]
  -R CELLSIZE           Regridding (nest) Factor. default=10
  -t THRESHOLD          Number of routing grid cells to define stream.
                        default=200
  -o OUT_ZIP_FILE       Output routing stack ZIP file
  -O OVROUGHRTFAC_VAL   OVROUGHRTFAC value. default=1.0
  -T RETDEPRTFAC_VAL    RETDEPRTFAC value. default=1.0
  --starts CHANNEL_STARTS
                        Path to channel initiation points feature class. Must
                        be 2D point type. [OPTIONAL]
  --gw GW_POLYS         Path to groundwater polygons feature class [OPTIONAL]

(wrfh_gis2) C:\Users\ksampson\Documents\GitHub\wrf_hydro_gis_preprocessor>
```

Inputs

- Required:
  - -i IN_GEOGRID - WRF GEOGRID file (.nc)
  - -d INDEM - High-resolution elevation raster file (Esri GRID, GeoTIFF, VRT, etc.)
  - -R CELLSIZE - Regridding Factor, or the nesting relationship of routing cells to LSM cells
  - -t THRESHOLD - Minimum basin size (in routing grid cells)
- Optional Parameters:
  - --CSV IN_CSV – Forecast points or gaging station locations (.csv)
  - -l IN_RESERVOIRS - Lake Polygons (polygon feature class or .shp)
  - -b BASIN_MASK – Mask channelgrid only to forecast basins?
  - -r RB_ROUTING – Build Muskingum/Muskingum-Cunge routing parameters
  - -o OUT_ZIP_FILE – output .zip archive containing tool outputs.
  - --gw GW_POLYS – Groundwater polygons to prescribe groundwater basin locations
  - --starts CHANNEL_STARTS – Override –t parameter & provide network headwater points.
  - -O OVROUGHRTFAC_VAL – constant
  - -T RETDEPRFAC_VAL – constant

Input: WRF Geogrid File

The purpose of the Geogrid file is to define the simulation domain and interpolate various static geographical datasets to the model grid.

- GEOGRID is used in the WRF-Hydro GIS Pre-processor to define the domain's coordinate reference system, extent, resolution, and certain 2D variables:
  - HGT_M (elevation)
  - LU_INDEX (landuse)
- Currently supported GEOGRID coordinate systems
  - MAP_PROJ = 1 (Lambert Conformal Conic)
  - MAP_PROJ = 3 (Mercator)
  - MAP_PROJ = 6 (Cylindrical Equidistant but NOT w/ rotated pole)
  - MAP_PROJ = 2 (Polar Stereographic)

GEOGRID: Projected Coordinate System

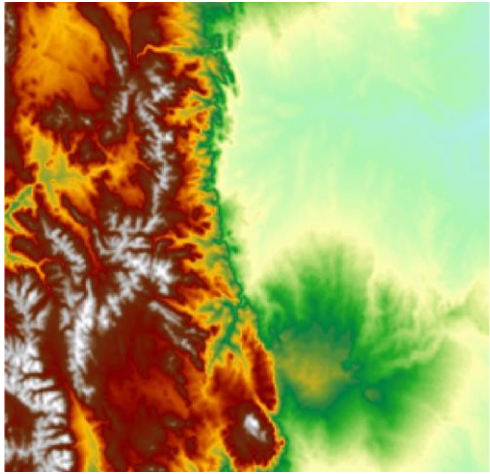
Front_Range_geo_em.d02.nc

```
:TITLE = "OUTPUT FROM GEOGRID V3.5.1";
:SIMULATION_START_DATE = "0000-00-00_00:00:00";
:WEST-EAST_GRID_DIMENSION = 50; // int
:SOUTH-NORTH_GRID_DIMENSION = 36; // int
:BOTTOM-TOP_GRID_DIMENSION = 0; // int
:WEST-EAST_PATCH_START_UNSTAG = 1; // int
:WEST-EAST_PATCH_END_UNSTAG = 49; // int
:WEST-EAST_PATCH_START_STAG = 1; // int
:WEST-EAST_PATCH_END_STAG = 50; // int
:SOUTH-NORTH_PATCH_START_UNSTAG = 1; // int
:SOUTH-NORTH_PATCH_END_UNSTAG = 35; // int
:SOUTH-NORTH_PATCH_START_STAG = 1; // int
:SOUTH-NORTH_PATCH_END_STAG = 36; // int
:GRIDTYPE = "C";
:DX = 1000.0f; // float
:DY = 1000.0f; // float
:DYN_OPT = 2; // int
:CEN_LAT = 39.940014f; // float
:CEN_LON = -105.42999f; // float
:TRUELAT1 = 30.0f; // float
:TRUELAT2 = 50.0f; // float
:MOAD_CEN_LAT = 39.940014f; // float
:STAND_LON = -105.0f; // float
:POLE_LAT = 90.0f; // float
:POLE_LON = 0.0f; // float
:corner_lats = 39.783337f, 40.093864f, 40.095993f, 3
:corner_lons = -105.714264f, -105.71753f, -105.14442
:MAP_PROJ = 1; // int
:MMINLU = "USGS";
:NUM_LAND_CAT = 24; // int
:ISWATER = 16; // int
:ISLAKE = -1; // int
:ISICE = 24; // int
:ISURBAN = 1; // int
:ISOILWATER = 14; // int
:grid_id = 1; // int
:parent_id = 1; // int
:i_parent_start = 1; // int
:j_parent_start = 1; // int
:i_parent_end = 50; // int
:j_parent_end = 36; // int
:parent_grid_ratio = 1; // int
:sr_x = 1; // int
:sr_y = 1; // int
:FLAG_MF_XY = 1; // int
}
```

WKT

```
"PROJCS['Lambert_Conformal_Conic',GEOGCS['GCS_Sphere',DATUM['D_Sphere',SPHEROID['Sphere',6370000.0,0.0]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Lambert_Conformal_Conic'],PARAMETER['false_easting',0.0],PARAMETER['false_northing',0.0],PARAMETER['central_meridian',-105.0],PARAMETER['standard_parallel_1',30.0],PARAMETER['standard_parallel_2',50.0],PARAMETER['latitude_of_origin',39.9400138855],UNIT['Meter',1.0]];
-36695400 -29251300 10000;-100000 10000;-100000
10000;0.001;0.001;0.001;IsHighPrecision"
```

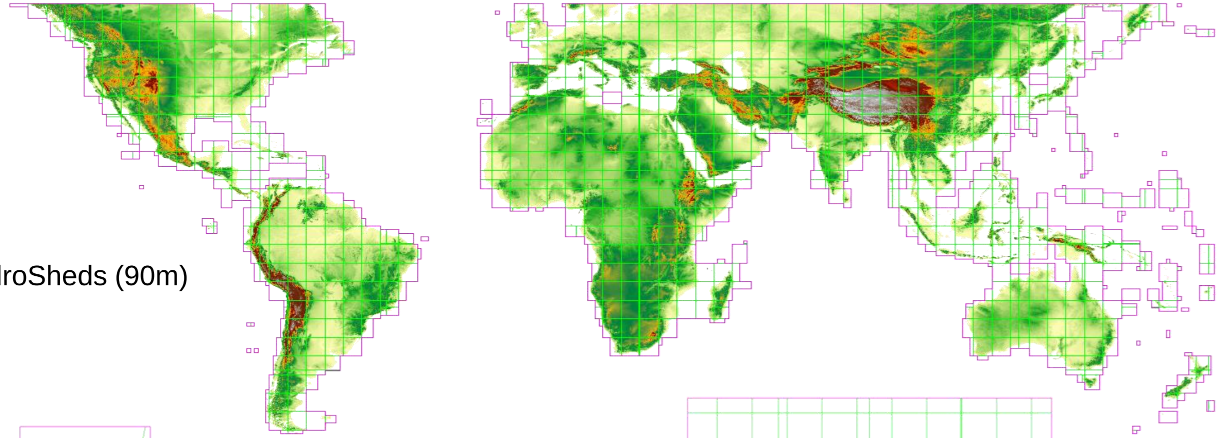
geo_em.d01.boulder_creek_1km.prj



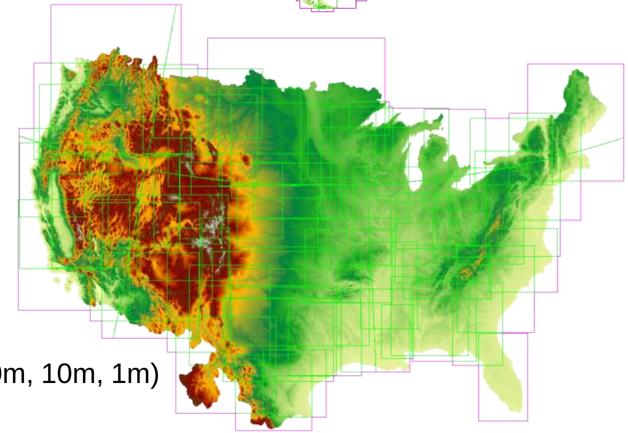
Input Elevation Raster

- Must be a GDAL-readable raster format
- A valid coordinate reference system is required
- Must cover entire extent (and more) of your domain
- Elevation units must be in meters ASL (m)
- Ideally elevation will be hydrologically corrected
  - Not necessary but helps with channel placement, hydro enforcement process.

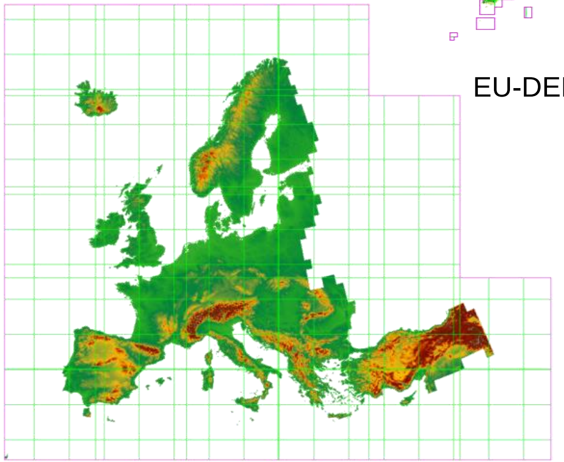
Input Elevation Data Sources



HydroSheds (90m)



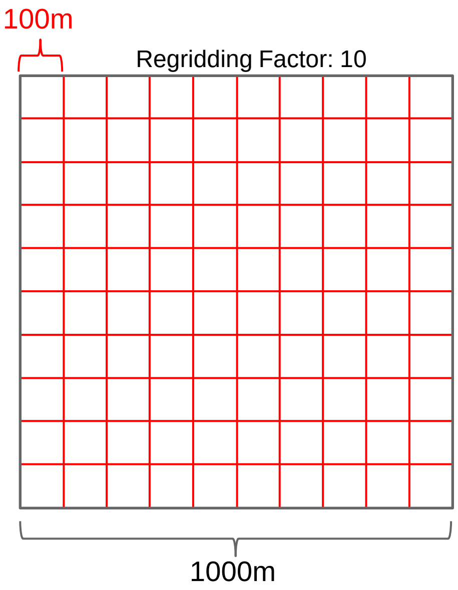
NED (30m, 10m, 1m)

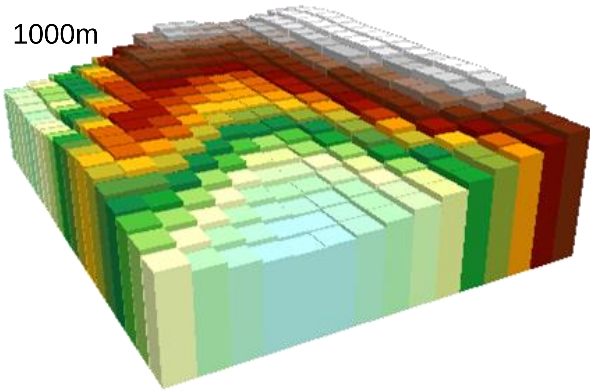


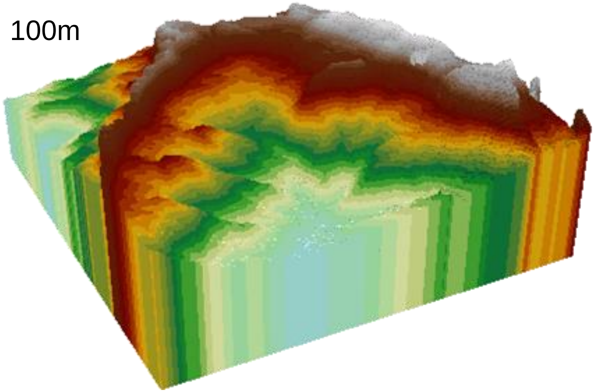
EU-DEM (25m)

Input Regridding Factor

$$\frac{\text{GEOGRID Resolution}}{\text{Regridding Factor}} = \text{Routing Resolution}$$

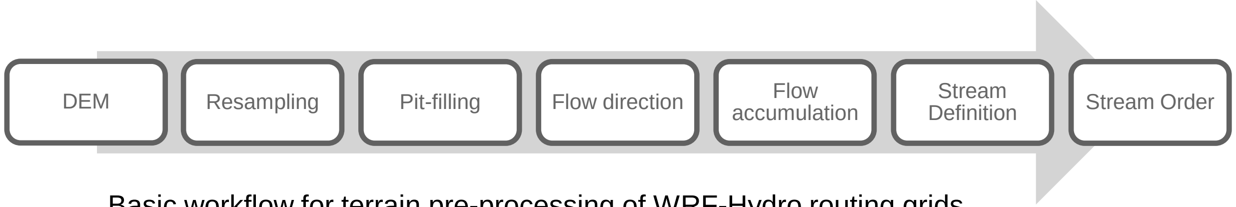






Terrain Pre-processing Workflow

- Resample high-resolution DEM
- Void-fill the resampled DEM
- D8 Flow Direction
- Flow Accumulation
- Derive CHANNELGRID from flow accumulation raster using basin area threshold or using channel initiation points
- Derive Strahler stream order from CHANNELGRID



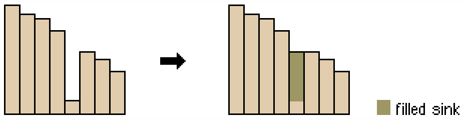
Basic workflow for terrain pre-processing of WRF-Hydro routing grids.

Reproject & Resample Hydro DEM

- Project input DEM to model projection and domain
- Resample to routing grid resolution
  - BILINEAR resampling uses a distance-weighted average of the 4 nearest cell centers.
- Re-projection and resampling can ‘break’ the hydrologic connectivity of the input HydroDEM.
  - Causing artificial ‘pits’.
  - Filling in ‘burned in’ areas.
  - Select a HydroDEM close to the desired routing resolution
- Even though we start with a HydroDEM, we ‘break’ it, then re-condition it.

Process: Pit Filling

Whitebox “Fill Depressions“ Tool



Profile view of a sink before and after running Fill

© Esri: http://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/how-fill-works.htm

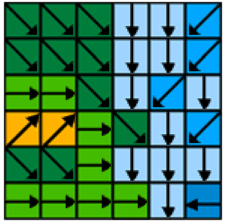
- Fill depressions so that water can roll downhill only. This also creates a smoother DEM than you might find in nature.
- This simple hydro-enforcement method can resolve most flow issues in a DEM.
- Optional z-limit (global variable) to limit fill depth.
  - Only compatible with Breach Depressions and Fill Depressions methods.

Flow Direction & Flow Accumulation

- D8 Flow Direction

32	64	128
16		1
8	4	2

- Flow Accumulation

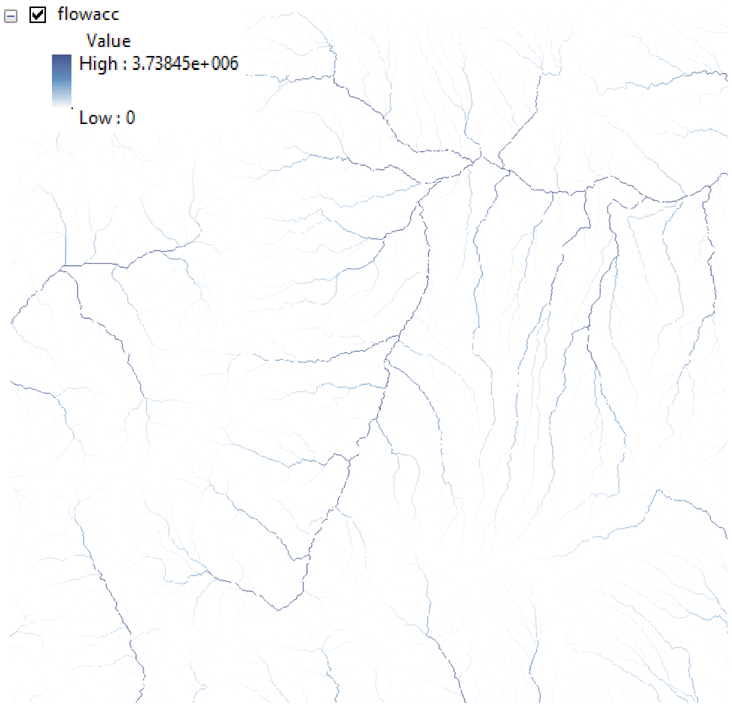


Flow direction

0	0	0	0	0	0
0	1	1	2	2	0
0	3	7	5	4	0
0	0	0	20	0	1
0	0	0	1	24	0
0	2	4	7	35	1

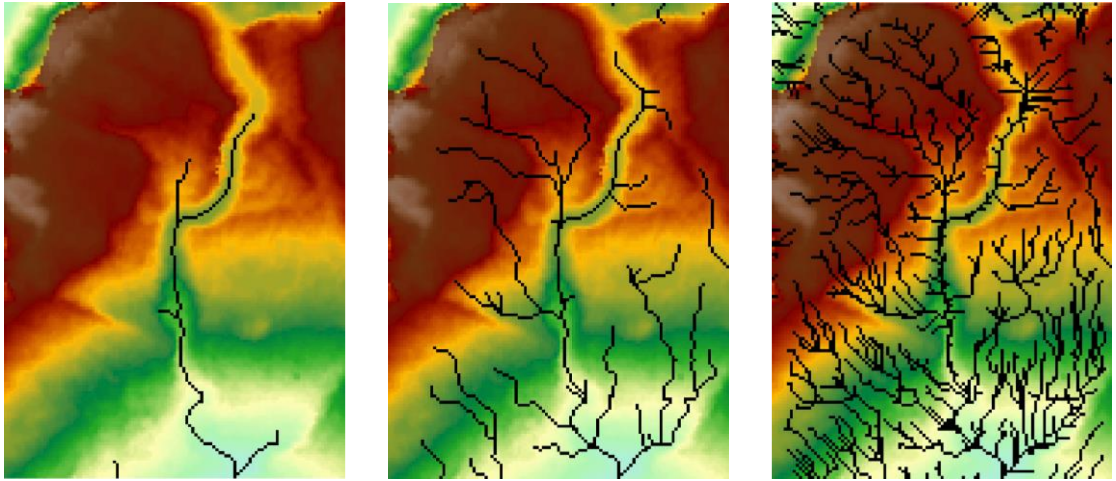
Flow accumulation

© Esri: http://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/how-flow-accumulation-works.htm



Stream Definition

- Input Parameter: Number of pixels to define stream
  - Yields a minimum ‘basin’ size
  - Given in pixels (unitless), on the routing grid
  - Affects density of generated channel network

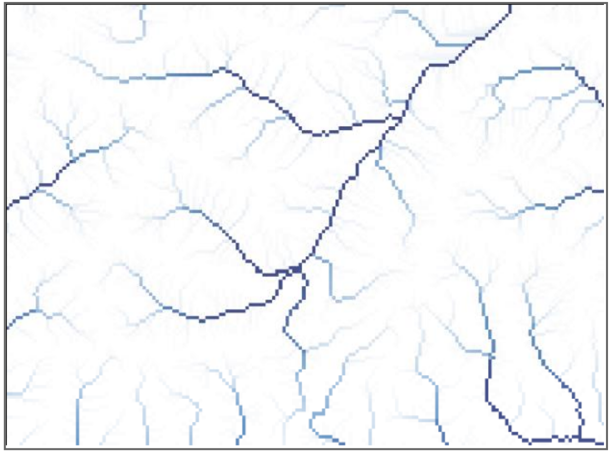


1km^2 0.1km^2 0.01km^2

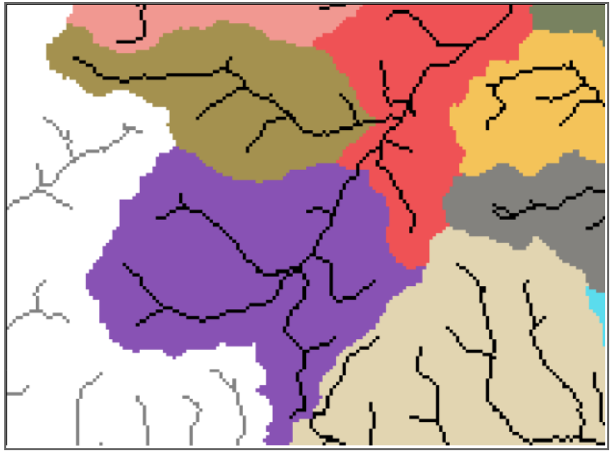
An analytic method for determining an appropriate threshold value for stream network delineation is presented in Tarboton et al. (1991)

Stream Definition

Flow Accumulation



Channelgrid



- Can use a threshold “–t” on flow accumulation to define channel cells
- Could use points as network-initiation locations (headwaters)
- Optional : use forecast basins as mask to activate channel cells
- If reach-based routing is selected, vector stream geometry is created
  - `streams.shp` shapefile written to output directory

Forecast Points (optional)

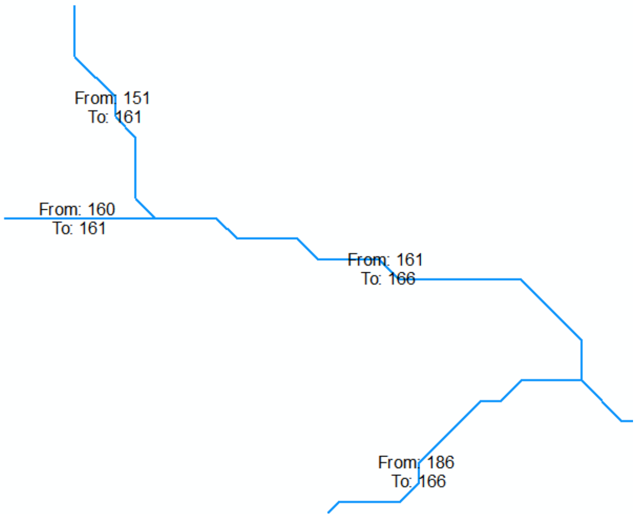
FID_	LON	LAT	STATION	Name	HOST	ELEV	DRAIN_AREA_SQMI	DRAIN_AREA_SQKM
1	-103.79889	40.26861	S_PLATTE_at_FT_MORGAN	6759500	USGS	4260	14627	37883.93
2	-108.26556	39.23917	COLO_at_CAMEO	9095500	USGS	4813	8050	20849.5
3	-104.39861	38.24806	ARKANSAS_nr_AVONDALE	7109500	USGS	4509	6327	16386.93
4	-105.88002	37.481392	RIO_GRANDE_nr_ALAMOSA	8223000	USGS	-9999	0	0

```
FID,LON,LAT,STATION,Name
15,-105.92833,40.08139,Fraser_at_Granby,9033300
18,-105.9,40.12083,COLO_nr_GRANBY,9019500
20,-106.3333,39.8803,Blue_R_blw_Grn_Mtn,9057500
```

- Create in Excel, Notepad, Word, etc.
- Direct output of attribute table from shapefile or feature class
- “LON”, “LAT”, “FID” required
- If present, basins will be delineated using the points provided
  - `frxst_basns` output variable will be created
  - `frxst_pts` & `basin_msk` variables will be populated
- If masked to basins, CHANNELGRID will have values -1, 0, -9999

Reach-Based Routing Background

- A vector-based approach to routing flow
- Channel network is comprised of ‘links’ instead of pixels
  - CHANNELGRID raster is converted to a line vector
  - LINKID grid written to Fulldom_hires.nc to store reach IDs
  - Elevation, latitude, longitude gathered at every node
- Muskingum-Cunge parameters applied to reaches
  - RouteLink.nc file stores necessary parameters such as length, slope, stream order, location, topology, etc.
- Computational efficiency vs. gridded methods



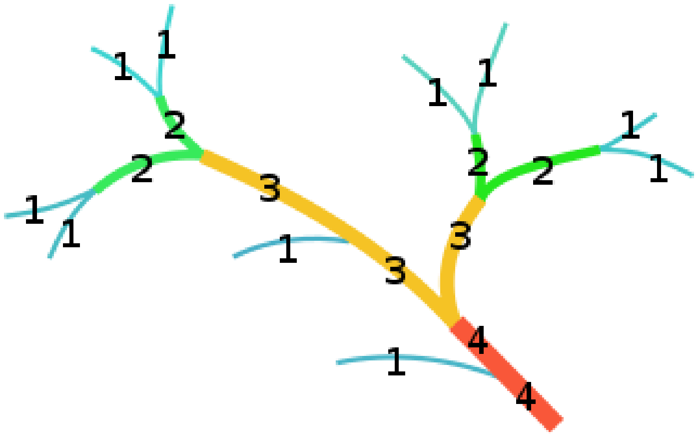
Reach-Routing Table

- CF-netCDF file containing reach-routing parameters
- Mix of derived attributes and default values

Parameter	Description
link	Link ID
from	From Link ID
to	To Link ID
lon	longitude of the start node
lat	latitude of the start node
alt	Elevation in meters from DEM at start node
order	Stream order (Strahler)
Qi	Initial flow in link (CMS)
MusK	Muskingum routing time (s)
MusX	Muskingum weighting coefficient
Length	Stream length (m)
n	Manning's roughness
So	Slope (meters/meter)
ChSlp	Channel side slope
BtmWdth	Bottom width of channel
Kchan	Channel conductivity (mm/hr)
x	X-coordinate in projected coordinate system
y	Y-coordinate in projected coordinate system

Process: Stream Order

- **Strahler Stream Order** Whitebox tool
  - Strahler (1957) method
  - Writes output file to STREAMORDER variable



Process: Reservoir Routing

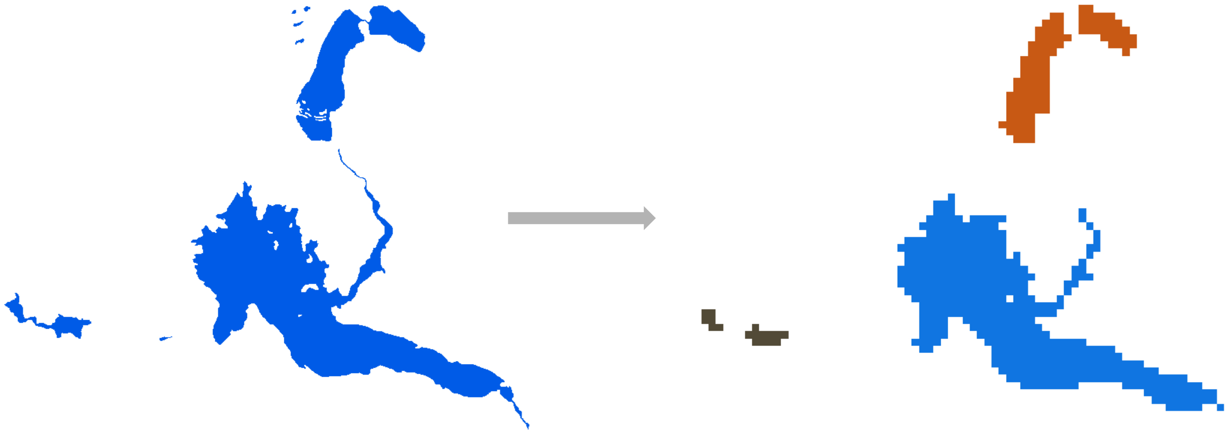
- If the option is selected, a polygon shapefile or feature class is required as input.
- Populates LAKEGRID variable
- Assigns lake ID values to pixels where lakes drain into channel
- Constructs a LAKEPARM.nc file with necessary variables for reach-based routing:
    - Lake area, max elevation, min elevation, base elevation, orifice elevation

```
netcdf LAKEPARM {
dimensions:
        nlakes = 82 ;
variables:
        int lake_id(nlakes) ;
                lake_id:long_name = "Lake ID" ;
                lake_id:cf_role = "timeseries_id" ;
        double LkArea(nlakes) ;
                LkArea:long_name = "Gridded lake area (sq. km)" ;
                LkArea:coordinates = "lat lon" ;
        double LkMxH(nlakes) ;
                LkMxH:long_name = "Maximum lake elevation (m ASL)" ;
                LkMxH:coordinates = "lat lon" ;
        double WeirC(nlakes) ;
                WeirC:long_name = "Weir coefficient" ;
                WeirC:coordinates = "lat lon" ;
        double WeirL(nlakes) ;
                WeirL:long_name = "Weir length (m)" ;
                WeirL:coordinates = "lat lon" ;
        double OrificeC(nlakes) ;
                OrificeC:long_name = "Orifice coefficient" ;
                OrificeC:coordinates = "lat lon" ;
        double OrificeA(nlakes) ;
                OrificeA:long_name = "Orifice cross-sectional area (sq. m)" ;
                OrificeA:coordinates = "lat lon" ;
        double OrificeE(nlakes) ;
                OrificeE:long_name = "Orifice elevation (m ASL)" ;
                OrificeE:coordinates = "lat lon" ;
        float lat(nlakes) ;
                lat:long_name = "latitude of the lake centroid" ;
                lat:units = "degrees_north" ;
                lat:standard_name = "latitude" ;
        float lon(nlakes) ;
                lon:long_name = "longitude of the lake centroid" ;
                lon:units = "degrees_east" ;
                lon:standard_name = "longitude" ;
        float alt(nlakes) ;
                alt:long_name = "vertical distance above mean sea level in (m ASL)" ;
                alt:standard_name = "height" ;
                alt:units = "m" ;
                alt:positive = "up" ;
                alt:axis = "Z" ;
        double time(nlakes) ;
                time:standard_name = "time" ;
                time:long_name = "time of measurement" ;
                time:units = "days since 2000-01-01 00:00:00" ;
        double Discharge(nlakes) ;
                Discharge:long_name = "Default Discharge" ;
                Discharge:units = "CMS" ;
                Discharge:coordinates = "lat lon" ;
        double WeirH(nlakes) ;
                WeirH:long_name = "Weir Height (m ASL)" ;
                WeirH:units = "m" ;
                WeirH:coordinates = "lat lon" ;

// global attributes:
                :featureType = "timeSeries" ;
                :history = "Created Mon May 02 14:43:24 2016" ;
}
```

LAKEGRID/LAKEPARM.nc

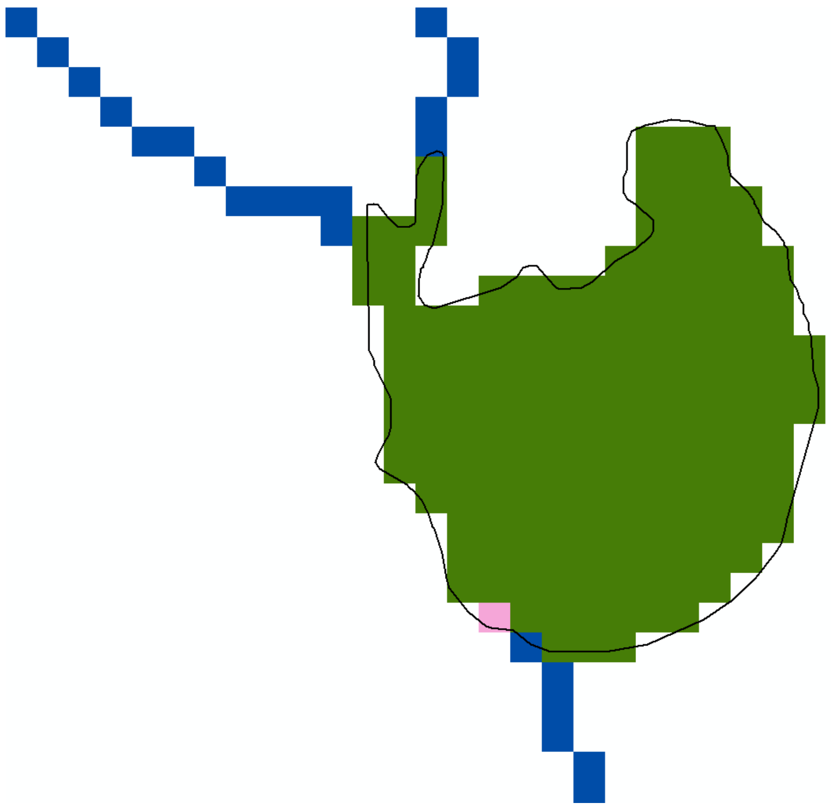
- Input: Reservoirs shapefile or feature class (polygon)
- Polygons are resolved on the model grid if they are large enough
- Lake ID is renumbered to 1-n



Original polygons

Lakes on the routing grid

Reservoir & Channels



Lakes/Reservoir Routing in the NWM

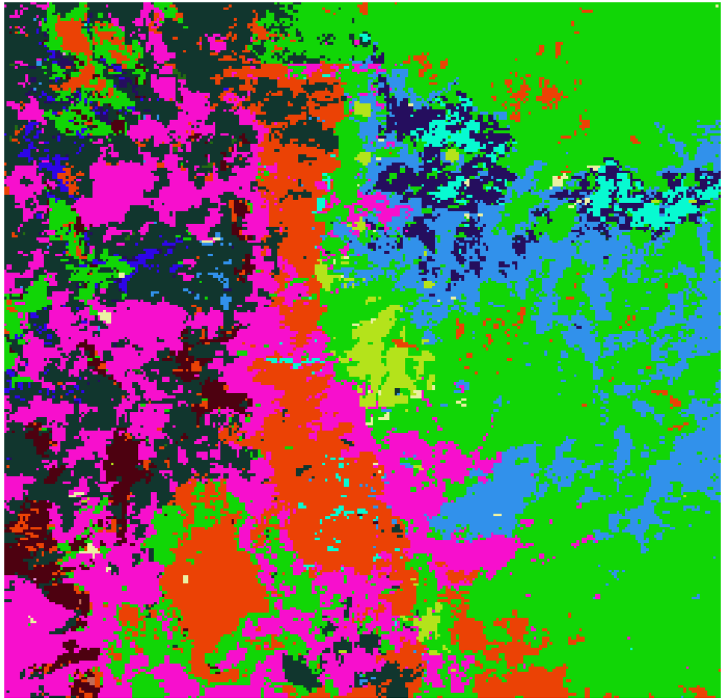
- Lake parameters are stored in the LAKEPARM.nc lake routing table
- Lakes are defined on the routing grid, parameters in the table

Lake Routing Table

Parameter	Description
LkArea	Gridded lake area (sq. km)
LkMxH	Maximum lake elevation (m ASL)
OrificeA	Orifice cross-sectional area (sq. m)
OrificeC	Orifice coefficient
OrificeE	Orifice elevation (m ASL)
WeirC	Weir coefficient
WeirH	Weir Height (m ASL)
WeirL	Weir length (m)
Ifd	Initial fractional water depth (% full)
lake_id	Lake ID
lat	latitude of the lake centroid
lon	longitude of the lake centroid
Dam_length	Dam length (m)

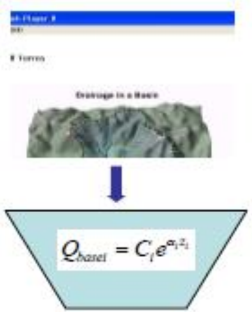
Other Grids

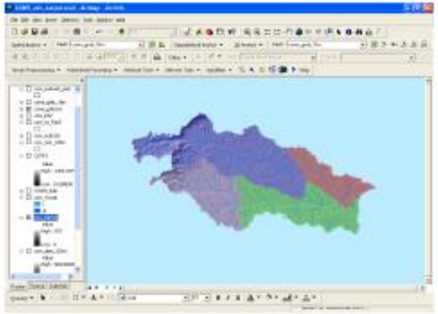
- Landuse
  - GEOGRID LU_INDEX resampled (nearest neighbor) to routing grid
- OVROUGHRTFAC
  - Constant 1.0 (float32)
- RETDEPRTFAC
  - Constant 1.0 (float32)
- LKSATFAC
  - Constant 1000.0 (float32)



Groundwater Buckets

- Conceptualized baseflow
- Spatially aggregated drainage from soil profile stored in ‘buckets’ representative of an aquifer
- GWBUCKPARM.nc bucket parameter file
- Buckets resolved on the coarse grid, written to a 2D netCDF file GWBASINS.nc





```
Basin,Coeff.,Expon.,Zmax,Zinit
        1,1.0000, 3.000, 150.00,10.0000
        2,1.0000, 3.000, 250.00,40.0000
        3,1.0000, 3.000, 150.00,30.0000
        4,1.0000, 3.000, 100.00,20.0000
        5,1.0000, 3.000, 100.00,50.0000
```

From WRF-Hydro User Guide, Figure 3.7

Groundwater Representation

- Groundwater Bucket Parameters
  - Built using default groundwater bucket parameters combined with LINKID-based local contributing basins.
  - Other methods available to produce groundwater basins from

    Forecast Points

    Polygon Shapefile

```
Select Windows PowerShell
PS C:\Data\Projects\Gochis\TOOLS\Standalone_Test_Data\Croton_Lambert\Croton_Lambert\Expected_Outputs\C
roton_r4_t32_reach> ncdump -h .\GWBUCKPARM.nc
netcdf GWBUCKPARM {
dimensions:
        feature_id = 38 ;
variables:
        int Basin(feature_id) ;
                Basin:long_name = "Basin monotonic ID (1...n)" ;
        float Coeff(feature_id) ;
                Coeff:long_name = "Coefficient" ;
        float Expon(feature_id) ;
                Expon:long_name = "Exponent" ;
        float Zmax(feature_id) ;
                Zmax:long_name = "Zmax" ;
        float Zinit(feature_id) ;
                Zinit:long_name = "Zinit" ;
        float Area_sqkm(feature_id) ;
                Area_sqkm:long_name = "Basin area in square kilometers" ;
        int ComID(feature_id) ;
                ComID:long_name = "Catchment Gridcode" ;

// global attributes:
                :featureType = "point" ;
                :history = "Created Tue Jun 05 14:10:30 2018" ;
}
```

GIS Pre-processor Outputs

- Set of netCDF, shapefile, ASCII & log files
  - 2-6 netCDF files
  - 0-1 ASCII Raster (.txt)
  - 0-2 Shapefiles
  - 1 .log file

- FRNG_1km_250m_Basic
  - Fulldom_hires.nc
  - GEOGRID_LDASOUT_Spatial_Metadata.nc

- FRNG_1km_250m_TestData_Full
  - Fulldom_hires.nc
  - GEOGRID_LDASOUT_Spatial_Metadata.nc
  - gw_basns_geogrid.txt
  - gw_basns_geogrid.prj
  - LAKEPARM.nc
  - lakes.shp
  - Route_Link.nc
  - streams.shp

Other Utilities

- Examine Outputs of GIS Preprocessor
  - Extracts .zip output file to individual rasters for viewing in Desktop GIS applications.
- Build GeoTiff from GEOGRID file
  - Export any M-grid variable from the GEOGRID file to raster format
- Export Projection Definition file (PRJ) from GEOGRID file
  - Reads GEOGRID file attributes and builds a projection file (.prj)
- Create Latitude and Longitude Rasters
  - Builds latitude and longitude grids from any raster input
- Create Domain Boundary Shapefile
  - Creates a polygon shapefile defining the domain boundary from a GEOGRID file
- Build Groundwater Inputs
  - Creates groundwater input files in 3 ways

Tool Messages

```
(wrfh_gis2) C:\Users\ksampson\Documents\GitHub\wrf_hydro_gis_preprocessor>python Build_Routing_Stack.py
Script initiated at Wed Oct 28 17:02:06 2020
usage: Build_Routing_Stack.py [-h] -i IN_GEOGRID [--CSV IN_CSV]
                              [-b BASIN_MASK] [-r RB_ROUTING]
                              [-l IN_RESERVOIRS] -d INDEM [-R CELLSIZE]
                              [-t THRESHOLD] [-o OUT_ZIP_FILE]
                              [-O OVROUGHRTFAC_VAL] [-T RETDEPRTFAC_VAL]
                              [--starts CHANNEL_STARTS] [--gw GW_POLYS]

This is a program to perform the full routing-stack GIS pre-processingfor WRF-
Hydro. The inputs will be related to the domain, the desired routing nest
factor, and other options and parameter values. The output will be a routing
stack zip file with WRF-Hydro domain and parameter files.

optional arguments:
  -h, --help            show this help message and exit
  -i IN_GEOGRID         Path to WPS geogrid (geo_em.d0*.nc) file [REQUIRED]
  --CSV IN_CSV          Path to input forecast point CSV file [OPTIONAL]
  -b BASIN_MASK         Mask CHANNELGRID variable to forecast basins?
                        [True/False]. default=False
  -r RB_ROUTING         Create reach-based routing (RouteLink) files?
                        [True/False]. default=False
  -l IN_RESERVOIRS      Path to reservoirs shapefile or feature class
                        [OPTIONAL]. If -l is TRUE, this is required.
  -d INDEM              Path to input high-resolution elevation raster
                        [REQUIRED]
  -R CELLSIZE           Regridding (nest) Factor. default=10
  -t THRESHOLD          Number of routing grid cells to define stream.
                        default=200
  -o OUT_ZIP_FILE       Output routing stack ZIP file
  -O OVROUGHRTFAC_VAL   OVROUGHRTFAC value. default=1.0
  -T RETDEPRTFAC_VAL    RETDEPRTFAC value. default=1.0
  --starts CHANNEL_STARTS
                        Path to channel initiation points feature class. Must
                        be 2D point type. [OPTIONAL]
  --gw GW_POLYS         Path to groundwater polygons feature class [OPTIONAL]
```

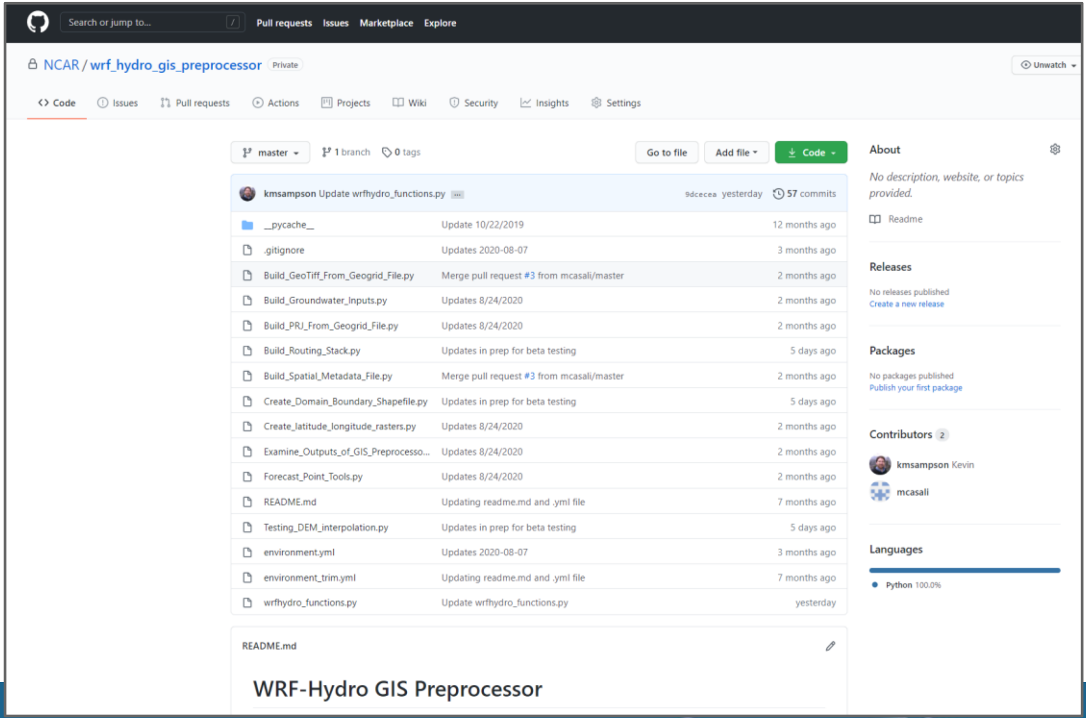
```
Script initiated at Wed Oct 28 21:52:05 2020
  Parameter values that have not been altered from script default values:
    Using default basin mask setting: False
    Using default reach-based routing setting: False
    Using default OVROUGHRTFAC parameter value: 1.0
    Using default RETDEPRTFAC parameter value: 1.0
  Values that will be used in building this routing stack:
    Input WPS Geogrid file: geo_em.d01.nc
    Forecast Point CSV file: C:\Users\ksampson\Desktop\CUAHSI_Nov_2020_Files\Test_Cases\Croton_Data\croton_frxst_pts_FOSS.csv
    Mask CHANNELGRID variable to forecast basins?: False
    Create reach-based routing (RouteLink) files?: False
    Lake polygon feature class: C:\Users\ksampson\Desktop\CUAHSI_Nov_2020_Files\Test_Cases\Croton_Data\lakes.shp
    Input high-resolution DEM: C:\Users\ksampson\Desktop\CUAHSI_Nov_2020_Files\Test_Cases\Croton_Data\NED_30m_Croton.tif
    Regridding factor: 4
    Stream initiation threshold: 20
    OVROUGHRTFAC parameter value: 1.0
    RETDEPRTFAC parameter value: 1.0
    Input channel initiation start point feature class: None
    Input groundwater basin polygons: None
    Output ZIP file: C:\Users\ksampson\Desktop\CUAHSI_Nov_2020_Files\Test_Cases\FOSS\example_case\Gridded\DOMAIN\Gridded_r4_t20_lak
es_frxst_mask.zip
  Running Process GEOGRID function
    Forecast points provided.
  Reach-based routing files will not be created.
WPS netCDF projection identification initiated...
    Map Projection: Lambert Conformal Conic
    Using MOAD_CEN_LAT for latitude of origin.
    Using Standard Parallel 2 in Lambert Conformal Conic map projection.
    Geo-referencing step completed without error in  0.11 seconds.
  Building sub-grid of model grid.
    Original grid spacing dx=1000.0, dy=-1000.0
    Original grid size: rows=16, cols=15
    New grid spacing: dx=250.0, dy=-250.0
    New dimensions: rows=64, cols=60
    Created projection definition from input NetCDF GEOGRID file.
    Proj4: +proj=lcc +lat_0=41.4710083007813 +lon_0=-97 +lat_1=30 +lat_2=60 +x_0=0 +y_0=0 +R=6370000 +units=m +no_defs
    Coarse grid GeoTransform: 1841998.7848425123 1000.0 0 278496.6013001092 0 -1000.0
    Coarse grid extent [Xmin, Ymin, Xmax, Ymax]: [1841998.7848425123, 262496.6013001092, 1856998.7848425123, 278496.6013001092]
    Fine grid extent [Xmin, Ymin, Xmax, Ymax]:   [1841998.7848425123, 262496.6013001092, 1856998.7848425123, 278496.6013001092]
    GDAL Data type derived from input array: 6 (float32)
  Creating CF-netCDF File.
    Esri PE String: PROJCS["unnamed",GEOGCS["GCS_Sphere",DATUM["D_Sphere",SPHEROID["unnamed",6370000.0,0.0]],PRIMEM["Greenwich",0.0
],UNIT["Degree",0.0174532925199433]],PROJECTION["Lambert_Conformal_Conic"],PARAMETER["False_Easting",0.0],PARAMETER["False_Northing
",0.0],PARAMETER["Central_Meridian",-97.0],PARAMETER["Standard_Parallel_1",30.0],PARAMETER["Standard_Parallel_2",60.0],PARAMETER["L
```

Documentation & Test Data

- README, Jupyter Notebook, and tool parameter help documentation currently available
- Detailed documentation forthcoming.
- Small GEOGRID domain for testing tool functionality
  - Croton (Lambert Conformal Conic)
- Required Elevation files (.tif) provided
- Optional stream gages & lakes provided
- Other optional inputs provided (groundwater basins, network start points)

GitHub

- https://github.com/NCAR/wrf_hydro_gis_preprocessor



Thank you

Questions: wrfhydro@ucar.edu

GIS Pre-processing Demostration