Lower Rio Grande Valley Development Council Flood Infrastructure Fund

Tier I Data Extraction for Tier II Modeling

Project Deliverable ID 1.2.1.4.3.2.1

Andrew N.S. Ernest, Ph.D., P.E., BCEE, D.WRE
and Christopher B. Fuller, Ph.D.
and William Kirkey, Ph.D.
and Peter Kirkey,
and Linda Navarro,
and Ivan Santos-Chavez

Aug 02, 2022

Approval Page

Technical Review By:

DocuSigned by:

William Kirkey

8/3/2022

CC59C65904A2440...

William Kirkey, Ph.D.

Chief of Research and Technology Development

Final Approval For Submission:

DocuSigned by:

8/2/2022

BC4713A8EA924AA...

Andrew N.S. Ernest, Ph.D., P.E., BCEE, D.WRE

Chief Executive Officer

August 2, 2022

CONTENTS

CHAPTER

ONE

ABOUT TIER I TO TIER II DATA EXTRACTION

This report describes the process of extraction of Tier I data for use within the Tier II modeling effort within the *RGVFlood.com* Cyberinfrastructure. This report is produced as part of *TWDB FIF* Deliverable ID 1.2.1.4.3.2.1.

CHAPTER

TWO

GOALS AND OBJECTIVES

The overarching goal of this work is to develop and deploy procedures for automatically collecting and extracting output data from *WRF-Hydro* for use as input and forcings for the *HEC-RAS*.

- Analyze and characterize the input and output files for the selected Tier I model.
- Evaluate available post-processing tools for extraction and ingestion of Tier I model data.
- Evaluate options for extraction of Tier I model data in real-time.
- Perform a preliminary assessment of the data conditioning needs for Tier I model output.

CHAPTER

THREE

WRF-HYDRO DATA FILES

The summaries provided below are adapted from WRF-Hydro StandAlone.

3.1 Input Files (Standalone Mode)

Geographical input data

This data, used by the *WRF WPS* `geogrid.exe`, is downloaded from UCAR WPS GEOG to define the domain's physical location on the globe.

Domain Coordinates

Domain Coordinates define the location of the domain and is used by the *WRF WPS* as input in the file 'namelist.wps' to produce GEOGRID files. These can be identified using the WRF Domain Wizard.

Digital Elevation Model/Elevation Data

Input data for the *WRF-Hydro GIS* Pre-processing Tool. *DEM* data can be obtained from:

* USGS HydroSHEDS
* National Hydrography Dataset Plus, Version 2
* EU-DEM
* Alternative Sources (NASA, etc)

Forecast Points (optional)

Latitude, Longitude and Station Name, used by the *WRF-Hydro GIS* Pre-processing Tool to specify locations for forecast production.

Lake Polygons (optional)

Also used by the *WRF-Hydro GIS* Pre-processing Tool to locate lakes and reservoirs.

Meteorological Forcing Data

Meteorological data for the simulation period to “force” the model simulation.

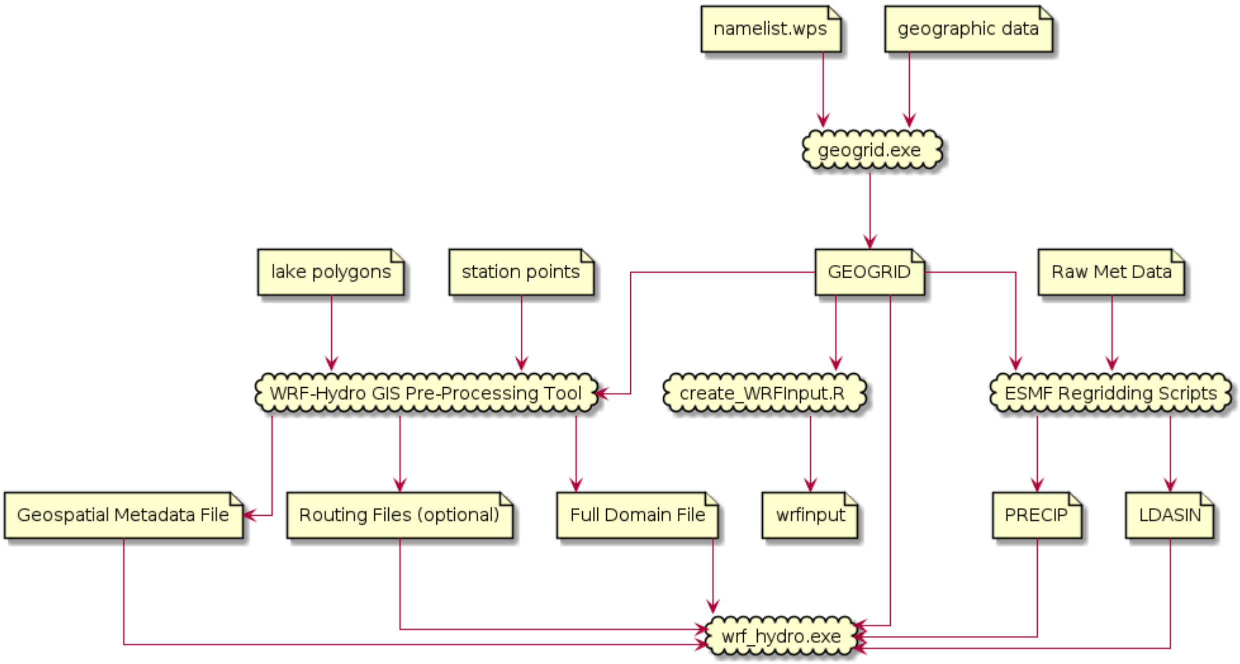


Figure3.1: WRF-Hydro Input Data Flow

3.2 Output Files (Standalone Mode)

The summaries provided below are adapted from WRF-Hydro Technical Description and User’s Guide

Standard WRF Output

Standard WRF model output, when run in coupled mode
wrfout”

LSM Gridded Output

LSM gridded output (netcdf)
YYYYMMDDHHMM.LDASOUT_DOMAINX
Model output on the land surface model grid written to a multi-dimensional netcdf data file.

Land Surface Diagnostic Output

YYYYMMDDHHMM.LSMOUT_DOMAINX

Channel Network Output

Full channel network output (netcdf point file)

YYYYMMDDHHMM.CHRTOUT_DOMAINX

A netcdf point file containing streamflow discharge, flow depth (i.e. 'stage' or 'head'), longitude, latitude, forecast point index and stream order value for all channel pixels within the high-resolution terrain data file.

Station Observations

Station observations (netcdf point file AND ascii Timeseries)

YYYYMMDDHHMM.CHANOBS_DOMAINX

frxst_pts_out.txt

A netcdf point file containing streamflow discharge, flow depth (i.e. 'stage' or 'head'), longitude, latitude, forecast point index and stream order value for each forecast point specified in the 'frxst_pts' data layer of the high-resolution terrain data file.

Streamflow on the 2D high resolution routing grid

YYYYMMDDHHMM.CHRTOUT_GRIDX

A netcdf point file containing streamflow discharge at each location in the grid.

Gridded Routing Output

High resoluIon gridded output (netcdf, not common due to filesize)

YYYYMMDDHHMM.RTOUT_DOMAINX

Channel Inflow

Channel-inflow (ascii Timeseries)

qstrmvolrt_accum.txt

Lake/Reservoir Output

Lake/reservoirvoutput (netcdf point file)

YYYYMMDDHHMM.LAKEOUT_DOMAINX

Groundwater/Baseflow

Groundwater/baseflow output (3 ascii Timeseries files)
YYYYMMDDHHMM.GWOUT_DOMAINX

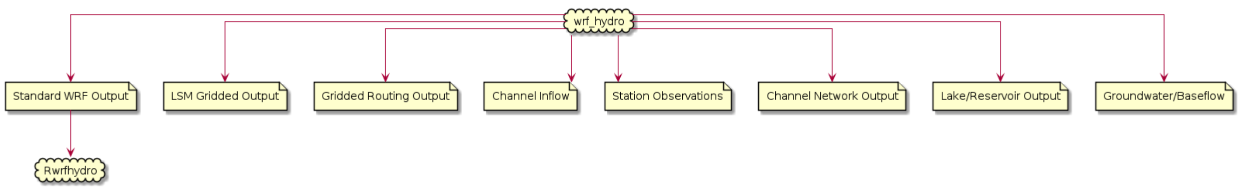


Figure3.2: WRF-Hydro Output Data Flow

3.3 NetCDF

NetCDF is the primary multi-dimensional data storage format for *WRF-Hydro* outputs. It is a fully transparent, open source format that has enabled a plethora of *FOSS* data and libraries for both creating and reading the files. The primary data outputs from *WRF-Hydro* of use to *HEC-RAS* models are the channel network output and the gridded streamflow output.

CHAPTER

FOUR

HEC-RAS DATA FILES

A HEC-RAS run requires three files: a Geometry datafile containing geometric features and data, a Steady Flow datafile containing flow rates and boundary conditions, and a Steady Flow Analysis (Plan) file, which contains the configuration that HEC-RAS uses to process the data (Figure 4). These files are created via four steps:

1. User-drawn features are created in RAS Mapper.
2. The geometry file is created based on the RAS Mapper contents
3. The Steady Flow datafile is created containing the flow rate at each cross-section in the geometry file, and boundary conditions for the geometric extents.
4. The Steady Flow Analysis is configured.

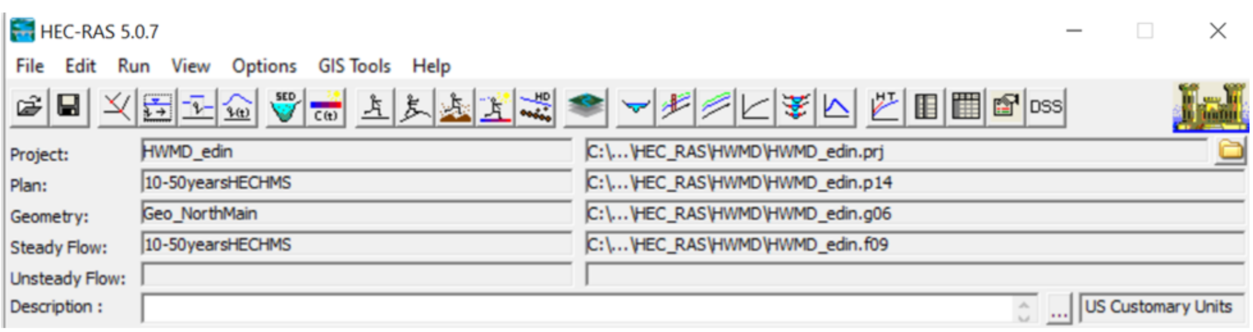


Figure4.1: HEC-RAS Input File Dialog

4.1 HEC Data Storage System

HEC-DSS

The U.S. Army Corps of Engineers' Hydrologic Engineering Center Data Storage System, or HEC-DSS, is a database system designed to efficiently store and retrieve scientific data that is typically sequential. Such data types include, but are not limited to, time series data, curve data, spatial-oriented gridded data, and others. The system was designed to make it easy for *users and application programs to retrieve and store data. HEC-DSS is incorporated into most of HEC's major application programs.*

pydsstools

pydsstools is an experimental Cython based Python library to manipulate HEC-DSS database file. It supports regular/irregular time-series, paired data series and spatial grid records. It is compatible with 64-bit Python on Windows 10 and Ubuntu like linux distributions. For the later, zlib, math, quadmath, and gfortran libraries must be installed. dssvue python library provides graphical user interface for HEC-DSS. There is also a Rust binding hecdss for HEC-DSS.

4.2 Ingestion Implications

Manual ingestion of CSV data into *HEC-RAS* is certainly possible. Exporting *WRF-Hydro* output data into CSV can be automated, however, the automated ingestion into *HEC-RAS* is uncertain. Considering the potential volume of discharge data that could be transferred from *WRF-Hydro* to *HEC-RAS*, the *HEC-DSS* file structure is likely the most efficient method, however, its format and structure is not readily available to be constructed from scratch. The existence of the *FOSS* pydsstools package is promising, however, more due diligence must be expended to ensure its feasibility.

CHAPTER

FIVE

POST PROCESSING TOOLS

The following post-processing tools were reviewed for potential data extraction application.

- Integrated Data Viewer by *Unidata*
- RWRFHydro by *NCAR*

The followung test data were used:

Croton, NY
Provided by *NCAR* as a *WRF-Hydro* test case.

LRGV, TX
Produced by *RATES* in the *RGVFlood* implementation strategy.

5.1 Integrated Data Viewer

IDV is a 3D geoscience visualization and analysis tool produced by *UniData*. It is a powerful visualization tool that focuses on the interactive experience and its ability to integrate multiple data sources and type (WRF-Hydro post processing vizualization).

5.2 Introduction to R

R is a programming language used for statistical computing and graphical presentation. It's primary potential use in this project is for the analysis & extraction of *WRF-Hydro* data and the creation of input data files for *HEC-RAS*

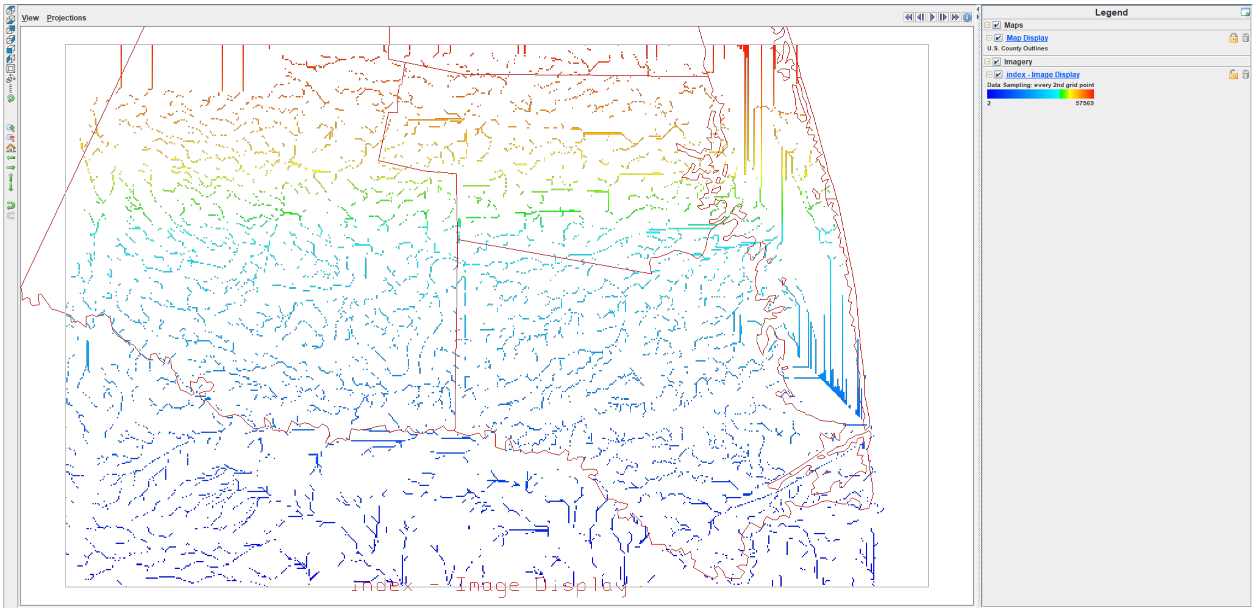


Figure5.1: LRGV Data Displayed in IDV

5.2.1 Installing on Ubuntu

```
#!/bin/bash
# https://cran.r-project.org/bin/linux/ubuntu/
# update indices
sudo apt update -qq
# install two helper packages we need
sudo apt install --no-install-recommends software-properties-common dirmngr
# add the signing key (by Michael Rutter) for these repos
# To verify key, run gpg --show-keys /etc/apt/trusted.gpg.d/cran_ubuntu_key.asc
# Fingerprint: E298A3A825C0D65DFD57CBB651716619E084DAB9
wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc | sudo tee -a /etc/apt/trusted.gpg.d/cran_ubuntu_key.asc
# add the R 4.0 repo from CRAN -- adjust 'focal' to 'groovy' or 'bionic' as needed
sudo add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu $(lsb_release -cs)-cran40/"
# Add the current R 4.0 or later c2d4u repository
sudo add-apt-repository ppa:c2d4u.team/c2d4u4.0+
# update indices
sudo apt update -qq
sudo apt install -y r-base r-cran-devtools
```

5.2.2 Basic Concepts

R shows significant potential as a key instrument for transforming *WRF-Hydro* output data into *HEC-RAS* input data. The basic definitions below highlight the key role *R* is expected to play with the *Python* based `Clover` package developed by *RATES* to manage cloud-based modeling systems.

Vignette
A vignette is a long-form guide to the package. It describes the type of problem it is designed to solve, and demonstrates solutions.

Variables

```
# Assignment using equal operator.
var1 = 3

# Assignment using leftward operator.
var2 <- "learn"

# Assignment using rightward operator.
TRUE -> var.3

print(var1)
cat ("var1 is ", var1 ,"\n")
cat ("var2 is ", var2 ,"\n")
cat ("var3 is ", var3 ,"\n")
```

Functions

```
my_function <- function(fname) {
paste(fname, "Griffin")
}

my_function("Peter")
my_function("Lois")
my_function("Stewie")
```

Lists

```
vec <- c(1,2,3)
char_vec <- c("Hadoop", "Spark", "Flink", "Mahout")
logic_vec <- c(TRUE, FALSE, TRUE, FALSE)
out_list <- list(vec, char_vec, logic_vec)
out_list
```

Dataframes

```
# Create a data frame
Data_Frame <- data.frame (
    Training = c("Strength", "Stamina", "Other"),
    Pulse = c(100, 150, 120),
    Duration = c(60, 30, 45)
)

# Print the data frame
Data_Frame
```

5.3 rwrfhydro

> *A community-contributed tool box for managing, analyzing, and visualizing WRF Hydro (and HydroDART) input and output files in R.* (An overview of rwrfhydro functionality)

rwrfhydro is an *R* package conceived by *NCAR* containing routines and libraries to analyze and visualize *WRF-Hydro* output data. It is named to sound like "*Our* WRF-Hydro". The routines packaged with rwrfhydro form a rigorous basis for integration with the *Clover* package. These routines are well documented in a series of Vignettes.

5.3.1 WRF Hydro Domain and Channel Visualization

An example of the the potential for combining functionality from rwrfhydro with *Clover* is in the use of the `VisualizeDomain` routine from rwrfhydro within *Clover*. `VisualizeDomain` creates plots of the WRF Hydro domain files using the rwrfhydro routine `VisualizeDomain` embedded in *Clover* using the `rpy2` package. The code block below is what would be used within rwrfhydro.

```
library("rwrfhydro")
VisualizeDomain(".../croton_NY/NWM/DOMAIN/Fulldom_hires.nc", "CHANNELGRID
↪")
```

5.3.2 visualize_domain

The `visualize_domain` method of the *Clover* package embeds the `VisualizeDomain` routine from `rwrfhydro` using the `rpy2` package:

`clover.fabfile.`**`visualize_domain`**(*geoFile='grid.nc'*, *pngFile='vizdom.png'*)

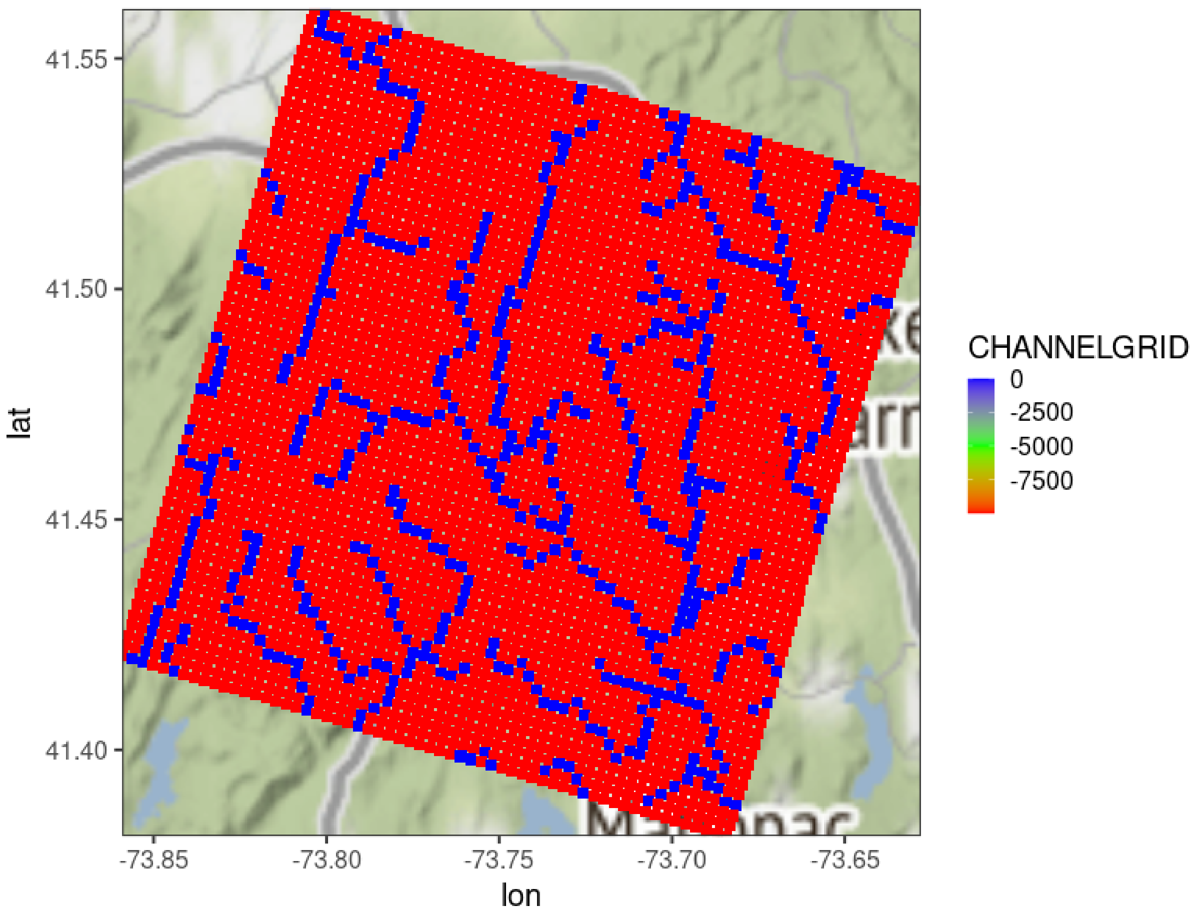
Plot the domain from the provided NetCDF file

Parameters

- **`geoFile`** (*str*) – NetCDF domain file path
- **`pngFile`** (*str*) – Output PNG file path

5.3.3 Croton, NY Domain

The *Clover* `visualize_domain` is used to visualize the `CHRTOUT_DOMAIN` output data from the Croton, NY test case.



CHAPTER

SIX

IMPLEMENTATION

6.1 Data Extraction Workflow

1. *WRF-Hydro* produces a CHRTOUT_GRID *NetCDF* file at predefined forecast times.
2. *Clover* reads the CHRTOUT_GRID file and extracts the `streamflow` gridded data.
3. For each available *HEC-RAS* dataset, *Clover* determines the boundary condition needs and populates the next forecast timestep in the appropriate *HEC-DSS* file.
4. When a user triggers a *HEC-RAS* dataset, the updated *HEC-DSS* file is provided.

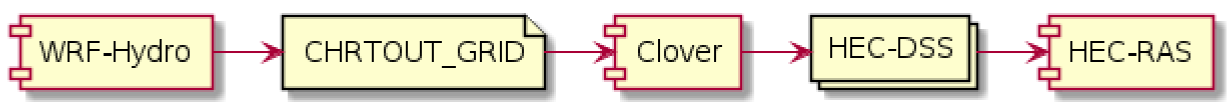


Figure6.1: Data Extraction Workflow

6.2 Test Data Used

Croton, NY
Provided by *NCAR* as a *WRF-Hydro* test case.

LRGV, TX
Produced by *RATES* in the *RGVFlood* implementation strategy.

6.3 Clover

The following sections describe modules provided by the *Clover* package. *Clover* is a comprehensive package that include modules that address *GCP* instantiation, model execution as well as data extraction and ingestion.

6.3.1 fabfile.py

Cloud Virtual Water Model Executioner, a framework for compiling and running selected water models on Google Cloud.

`clover.fabfile.`**`archive_to_bucket`**`(`*run_folder*, *bucket_name*, *name*`)`

Archive a folder to the bucket and delete it

Parameters

- **`run_folder`** (*str*) – Folder to archive
- **`bucket_name`** (*str*) – The bucket to archive to
- **`name`** (*str*) – The VM

`clover.fabfile.`**`create_godaddy_ip`**`(`*vmname='ross'*`)`

Create GoDaddy Domain IP

Parameters
`vmname` (*str*) – vm name

`clover.fabfile.`**`create_instance`**`(`*vmname='ross'*`)`

Create VM Instances

Parameters
`name` (*str*) – Virtual Machine instance listed in the config file. Also the VM name

Adapted from: GCP Create Instance

`clover.fabfile.`**`cxn`**`(`*vmname='ross'*`)`

Create the connection for the VM

Parameters
`vm` (*str*) – Virtual Machine defined in the configuration file

Return cxn
Connection object

Rtype cxn
object

`clover.fabfile.delete_instance(vmname='ross')`

Delete VM Instances

Parameters
vmname (*str*) – Virtual Machine instance listed in configuration file

`clover.fabfile.get_instance(vmname='ross')`

Get VM Instance

Parameters
vmname – Virtual Machine instance listed in the config file. Also the VM name

Return instance
Dictionary describing the instance

Rtype instance
dict

Adapted from: GCP Get Instances

`clover.fabfile.hecras_files(working_folder)`

Analyze the avalaible HEC-RAS data files and determine possible applications

Parameters
working_folder (*str*) – Working folder on the model host

Return usable_binary_combinations
Dictionary of lists of plan files useable for each of the Linux Binaries

Rtype usable_binary_combinations
dict

1. Copy the HEC-RAS dataset to the working folder on the model host
2. Check the plan files to make sure the appropriate combinations are available for each of the Geometry Pre-Processorm Unsteady and Steady flow executables
3. Saves the dictionary of usable combinations as a run.toml file in the working directory

`clover.fabfile.instance_ip(vmname='ross')`

Get VM Instance IP

Parameters
vmname (*str*) – Virtual Machine instance listed in the config file. Also the VM name

Return ip
IP Address

Rtype ip
str, IP Address as a string

Adapted from: GCP Get Instances

`clover.fabfile.list_godaddy_domains()`

List GoDaddy Domains

Godaddy_account
Goddady Account section listed in the configuration file

`clover.fabfile.load_data(model='HEC-RAS', dataset='Muncie')`

Loads the data onto the model host for ingestion

Parameters

- **model** (*str*) – Model for which the dataset is being located and loaded
- **dataset** (*str*) – The dataset to be loaded
- **datahost** (*str*) – Host where the data resides
- **modelhost** (*str*) – Host where the model resides

Return working_folder
The working folder for remaining operations

Rtype working_folder
str

1. Locate the dataset host
2. Locate the model host
3. Copy the dataset to the working folder on the model host
4. Returns the working folder on the model host

`clover.fabfile.map_grid(fn='grid.nc', pfn='mapgrid.png')`

Plot a WRF-Hydro NetCDF Domain or GRID file and overly on a Basemap. Adapted from Water Programming: A Collaborative Research Blog

`clover.fabfile.nc_hist(fn='grid.nc', m=0, pfn='nchist.png')`

Plot the streamflow data from a WRF-Hydro NetCDF Domain or GRID file.

`clover.fabfile.nc_hydrograph(dn='/home/anernest/Git/DataExtractor/downloads/WRFHydro_V2_CHRT`
`rn=1, lat=None, lon=None, pfn='hydro.png')`

Read the streamflow data over time from the GRID files in the gdn directory and plot the hydrograph

`clover.fabfile.nc_outlier_analysis(fn='grid.nc')` `clover.fabfile.`**`nc_stats`**`(`*fn='grid.nc'*, *m=2*`)`

Generate the statistics from a WRF-Hydro NetCDF file

`clover.fabfile.`**`ras_geompre`**`(`*working_folder*`)`

Run the RAS geometry pre-processor. From HEC-RAS User Manual (Geometry Preprocessor):

> The Geometric Preprocessor is used to process the geometric data into a series of hydraulic properties tables, rating curves, and family of rating curves. This is done in order to speed up the unsteady flow calculations. Instead of calculating hydraulic variables for each cross-section, during each iteration, the program interpolates the hydraulic variables from the tables. **The preprocessor must be executed at least once, but then only needs to be re-executed if something in the geometric data has changed.**

Required Files:

- Unsteady Flow Run File (.x??)
- Geometry HDF File (.g??.tmp.hdf)

Parameters

- **vmname** (*str*) – Virtual Machine instance listed in the config file. Also the VM name
- **basin** (*str*) – HEC-RAS Project Name
- **plan** (*str*) – Plan Number

1. Connect to the VM and read the run.toml file
2. Reads the run.toml file for instructions
3. Operates from the timestamped working folder on the specific VM
4. Deletes old outputs
5. Runs the RAS Geometry preprocessor on the Muncie file
6. Copies the output so it can be used by other Runs
7. Archives the run folder to the GCS bucket.

`clover.fabfile.`**`ras_steady`**`(`*basin='Muncie'*, *plan='04'*, *name='ross'*`)`

Run the Muncie Steady Flow test case. From HEC-RAS User Manual:

Parameters

- **name** (*str*) – Virtual Machine instance listed in the config file. Also the VM name
- **basin** (*str*) – HEC-RAS Project Name

- **plan** (*str*) – Plan Number

1. Creates a timestamped test case run folder
2. Copies the Muncie test case to the run folder
3. Deletes old outputs
4. Runs the RAS Steady on the Muncie file
5. Copies the output so it can be used by other Runs
6. Archives the run folder to the GCS bucket.

`clover.fabfile.ras_unsteady(working_folder)`

Run RAS Unsteady Flow. From HEC-RAS User Manual (Unsteady Flow):

> The unsteady flow computational program in HEC-RAS uses the same hydraulic calculations (cross section properties, bridge and culvert hydraulics, weirs, gated structures, etc…) that HEC developed for steady flow, however, the solution of the unsteady flow equations (Continuity and momentum equation) are solved using a unique skyline matrix solver developed by Dr. Robert Barkau for his UNET (Unsteady NETwork model) program. The unsteady flow simulation is actually a three-step process. First data is read from HEC-DSS, if necessary, and then converted into the user specified computation interval. Next, the RasUnsteady.exe program runs. This software reads the hydraulic properties tables computed by the pre-processor, as well as the boundary conditions and flow data from the interface. The program then performs the unsteady flow calculations. The final step is a program called RasDSSWriter.exe. This software takes the results from the RasUnsteady.exe run and writes the stage and flow hydrographs to an HEC-DSS file.

Parameters

- **name** (*str*) – Virtual Machine instance listed in the config file. Also the VM name
- **basin** (*str*) – HEC-RAS Project Name
- **plan** (*str*) – Plan Number

1. Connect to the VM and read the run.toml file
2. Reads the run.toml file for instructions
3. Operates from the timestamped working folder on the specific VM
4. Deletes old outputs
5. Runs the RAS UnSteady on the Muncie file

6. Copies the output so it can be used by other Runs
7. Archives the run folder to the GCS bucket.

`clover.fabfile.`**`read_nc`**`(`*fn='grid.nc'*`)`

Read NetCDF Files and display metadata. Adapted from Read NetCDF Data with Python

`clover.fabfile.`**`remove_hdf5_results`**`(`*filename*`)`

Remove the Results from an HDF5 file

Adapted from HEC-RAS Linux Download.

> *The file "remove_HDF_Results.py" is a python script for trimming the "Results" data group from the plan HDF file. When an unsteady flow compute is performed in the HEC-RAS GUI v.5.0.7, model information and computed results are stored in the plan HDF file, e.g LeveeBreach.p01.hdf. For the unsteady RAS compute in the CHPS, a *.p01.tmp.hdf is required. The file can be produced by stripping the "Results" data group from the *.p01.hdf file output from the GUI compute and renaming. A simple renaming of the *.p01.hdf to *.p01.tmp.hdf is not sufficient as the rasUnsteady64 program checks for the presence of the "Results" data group and exits if detected.*
>
> *The "Results" data group can be removed from the plan hdf file using a small Python program:*
>
> C:>python remove_HDF5_Results.py LeveeBreach.p01.hdf
>
> *The output of the script is the file, LeveeBreach.p01.tmp.hdf which will become one of the HEC-RAS model files in the Config/ModuleDataSet for the HEC-RAS river system.*
>
> *In the v.5.0.3 GUI, the creation of the *.p01.tmp.hdf file could be accomplished by only running the Geometry Preprocessor step in the Unsteady Flow Analysis. This process no longer available in the v.5.0.7 GUI.*

Parameters
`filename` (*str*) – HDF file to modify

`clover.fabfile.`**`reserve_static_ip`**`(`*vmname='ross'*`)`

Reserves a static IP

Parameters
`project` (*str*) – Project ID

Adapted from GCP Reserve IP

`clover.fabfile.`**`run_wrfhydro`**`(`*run='CROTON'*`)`

Runs a WRF-HYDRO scenario defined in config.ini. Defaults to the CROTON example case if not specified otherwise.

Parameters
run (*str*) – WRF-Hydro scenario to run, defined by the section name of the config.ini read

1. Defines the connection to VM_NAME
2. Downloads and decompresses the example case
3. Copies files from the WRF-Hydro source code to the run folder
4. Copies the data filees from the example case to the run folder
5. Runs the NoahMP executable
6. Archives the run folder

`clover.fabfile.`**`timestamp`**`()`

Return a formated timestamp for logging events

`clover.fabfile.`**`update_godaddy_ip`**`(`*domain*, *ip*`)`

Update GoDaddy Domain IP

Parameters

- **domain** (*str*) – Domain name
- **ip** (*str*) – IP address

`clover.fabfile.`**`visualize_domain`**`(`*geoFile='grid.nc'*, *pngFile='vizdom.png'*`)`

Plot the domain from the provided NetCDF file

Parameters

- **geoFile** (*str*) – NetCDF domain file path
- **pngFile** (*str*) – Output PNG file path

6.4 Croton NY

The Croton, NY output data was read using the *Clover* `read_nc` module to read and display metadata from `CHRTOUT_DOMAIN` output files.

```
201109020000.CHRTOUT_DOMAIN1
<class 'netCDF4._netCDF4.Dataset'>
root group (NETCDF4 data model, file format HDF5):
    featureType: timeSeries
    proj4: +proj=lcc +units=m +a=6370000.0 +b=6370000.0 +lat_1=30.0 +lat_
↪2=60.0 +lat_0=40.0 +lon_0=-97.0 +x_0=0 +y_0=0 +k_0=1.0 +nadgrids=@
```

(continues on next page)

(continued from previous page)

```
    model_initialization_time: 2011-08-26_00:00:00
    station_dimension: feature_id
    model_output_valid_time: 2011-09-02_00:00:00
    model_total_valid_times: 168
    stream_order_output: 1
    cdm_datatype: Station
    Conventions: CF-1.6
    code_version: v5.1.1-beta
    model_output_type: channel_rt
    model_configuration: retrospective
    dev_OVRTSWCRT: 1
    dev_NOAH_TIMESTEP: 3600
    dev_channel_only: 0
    dev_channelBucket_only: 0
    dev: dev_ prefix indicates development/internal meta data
    dimensions(sizes): feature_id(185), time(1), reference_time(1)
    variables(dimensions): int32 time(time), int32 reference_
↪time(reference_time), |S1 crs(), int32 feature_id(feature_id), float32␣
↪latitude(feature_id), float32 longitude(feature_id), int32␣
↪order(feature_id), float32 elevation(feature_id), float32␣
↪streamflow(feature_id), float32 nudge(feature_id), float32 q_
↪lateral(feature_id), float32 velocity(feature_id), float32␣
↪qSfcLatRunoff(feature_id), float32 qBucket(feature_id), float32␣
↪qBtmVertRunoff(feature_id)
    groups:
```

6.4.1 Croton Map-Grid

The Croton, NY `CHRTOUT_DOMAIN` *NetCDF* datafile was then read using the *Clover* `map_grid` module and displayed in the figure below.

```
/home/anernest/Git/DataExtractor/.venv/lib/python3.8/site-packages/
↪matplotlib/collections.py:981: RuntimeWarning: invalid value␣
↪encountered in sqrt
  scale = np.sqrt(self._sizes) * dpi / 72.0 * self._factor
Domain File 201109020000.CHRTOUT_DOMAIN1
```

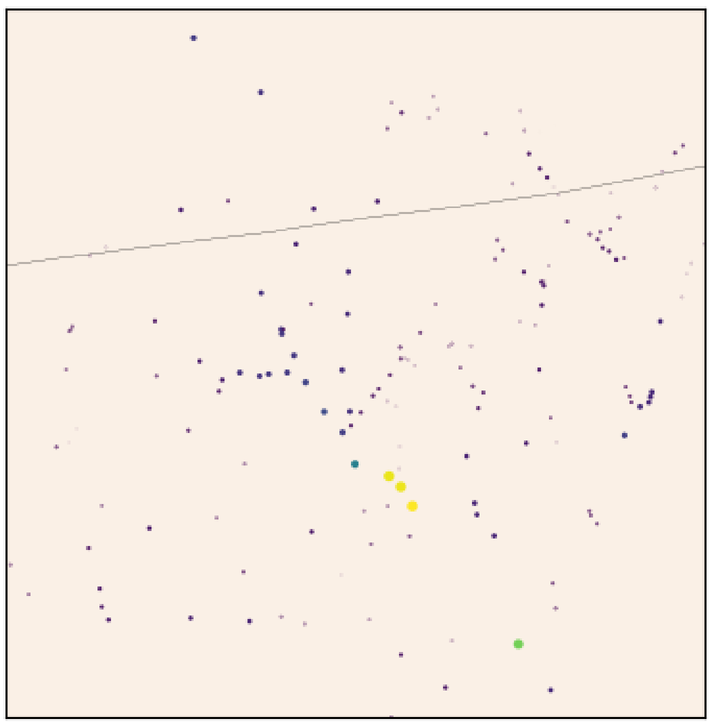


Figure6.2: Croton, NY Stream Network. Sparcity of points reflect the domain stream locations.

6.4.2 Croton Outlier Analysis

An outlier analysis was conducted on the same dataset. The table below depicts the output of the `nc_outlier_analysis` module. This module re-calculates the dataset simple statistics after filtering data that falls outside the specified number (`m`) of standard deviations, in ths case ranging from unfiltered (triggered by `m=0`) and 4. The generally accepted filter of 2 standard deviations (`m=2`) appears function well, indicating that the dataset is reasonably devoid of significant outliers.

..	min	max	mean	stdev
0	1.62033e-37	8.43465	0.538726	1.25101
1	1.62033e-37	1.69901	0.330557	0.419847
2	1.62033e-37	1.92247	0.340323	0.436626
3	1.62033e-37	3.62963	0.36038	0.505025
4	1.62033e-37	3.62963	0.36038	0.505025

6.4.3 Croton Streamflow Distribution (m=0)

The unfiltered streamflow is plotted in the chart below. While the bulk of the streamflow is in the 0-2 cms range, the maximum predicted flow is well within one order of magnitude of the mean flow. This is a reasonable indication that the data is representative of the physical system, and could be ingested as boundary conditions to a *HEC-RAS* model without fear of numerical instability.

6.4.4 Croton Streamflow Distribution (m=2)

The streamflow, filtered within 2 standard deviations, is plotted in the chart below, with all of the streamflow now in the 0-2 cms range. It is, however, likely that the filtered streamflows between 2 and 8 cms are reflective of the physical system and removal of them from the imposed boundary conditions in *HEC-RAS* will negatively affect the accuracy of the hydraulic model output.

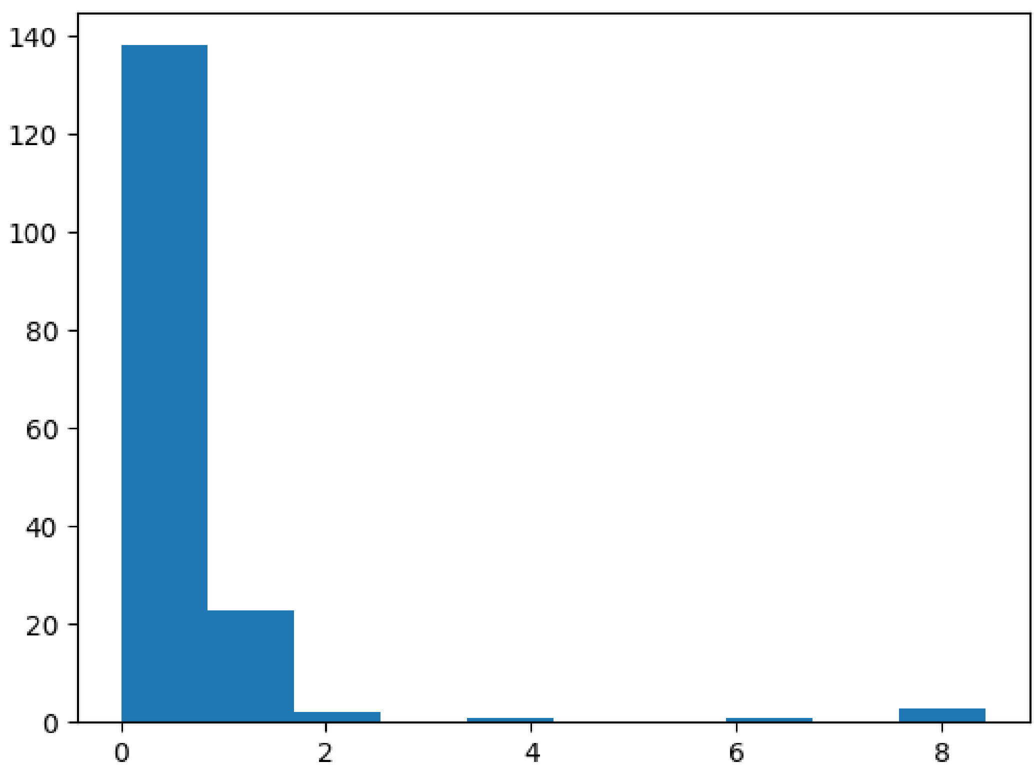


Figure6.3: Croton, NY Unfiltered Streamflow Distribution

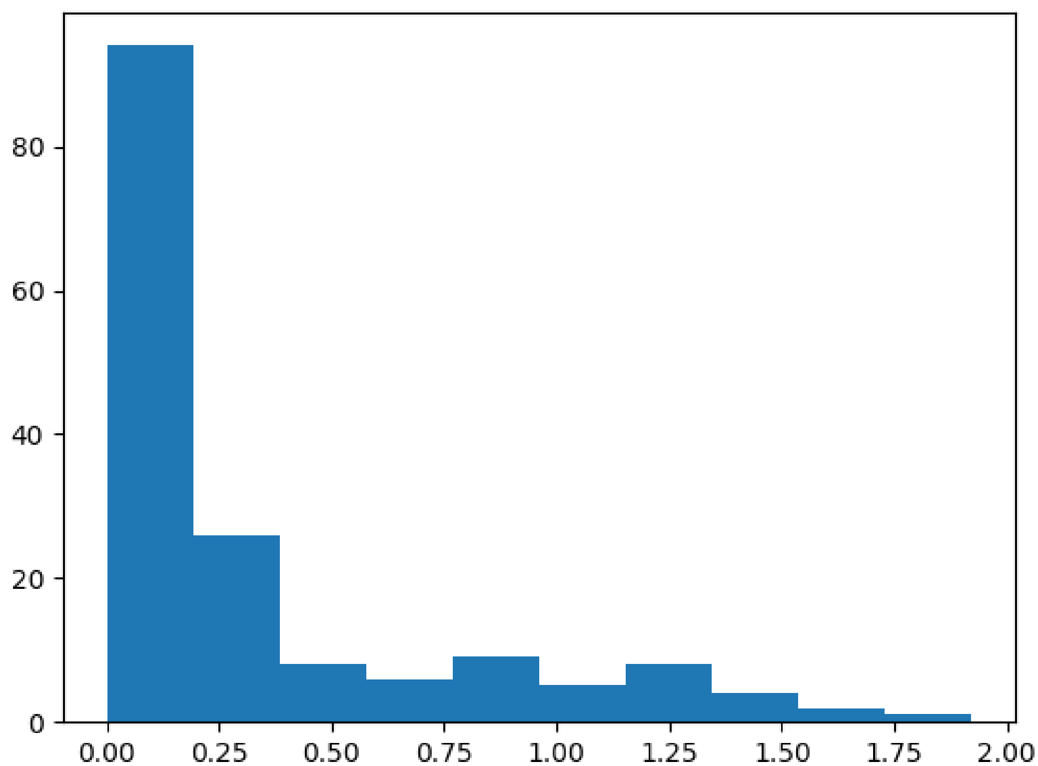


Figure6.4: Croton, NY Streamflow Distribution Filtered within 2 Standard Deviations

6.4.5 Croton Hydrograph

The Croton, NY was then plotted against time to produce the hydrograph below. Although the *NetCDF* specifications allow for multi-dimensional data storage, permitting multiple timesteps on the latitude-longitude domain to be stored within one file, this run produces a separate *NetCDF* file for each timestep, likely to account for potential filesystem file-size limitations. The composite data was produced by compiling the data across timesteps, specifying the location in the latitude-longitude domaim. In operational mode, these locations would be defined by the *HEC-RAS* boundary condition locations, and/or the locations of the *RTHS* stations.

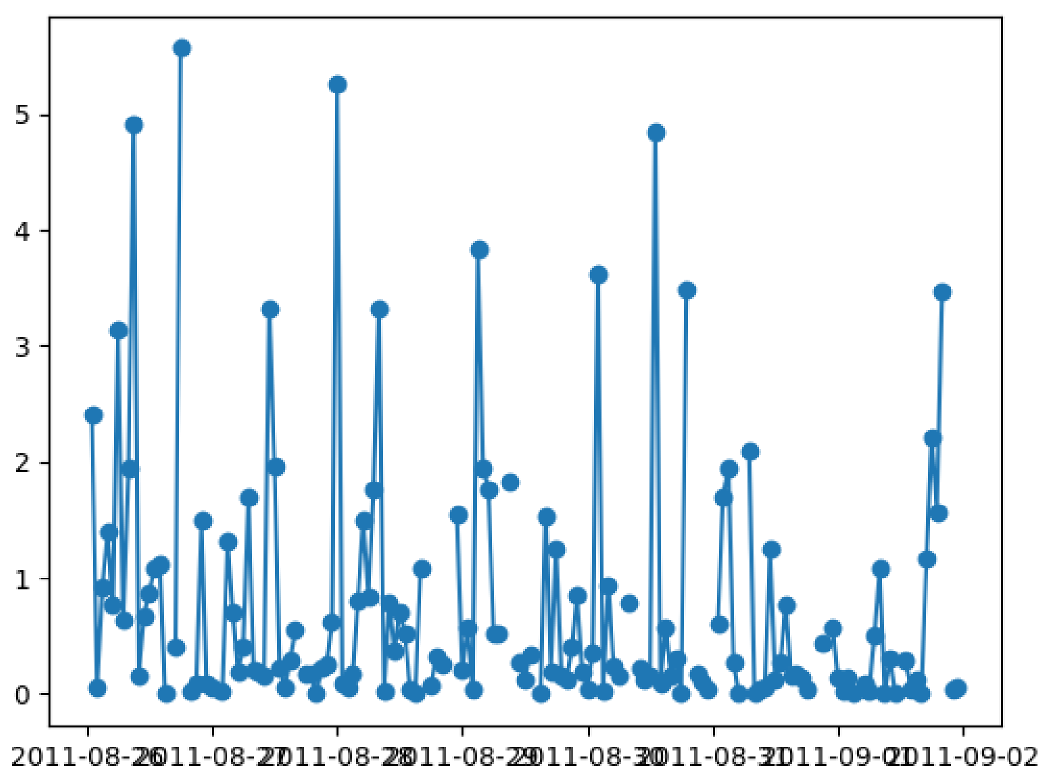


Figure6.5: Croton, NY Hydrograph

6.5 LRGV TX

The LRGV, TX output data was read using the *Clover* `read_nc` module to read and display metadata from `CHRTOUT_GRID` output files.

```
202111011300.CHRTOUT_GRID1
<class 'netCDF4._netCDF4.Dataset'>
root group (NETCDF4 data model, file format HDF5):
    TITLE: OUTPUT FROM WRF-Hydro v5.2.0
    model_initialization_time: 2021-11-01_00:00:00
    model_output_valid_time: 2021-11-01_13:00:00
    model_total_valid_times: 2160
    Conventions: CF-1.6
    code_version: v5.2.0
    model_output_type: channel_rt
    model_configuration: default
    proj4: +proj=lcc +units=m +a=6370000.0 +b=6370000.0 +lat_1=20.0 +lat_
↪2=60.0 +lat_0=26.235 +lon_0=-96.0 +x_0=0 +y_0=0 +k_0=1.0␣
↪+nadgrids=@null +wktext  +no_defs
    GDAL_DataType: Generic
    dimensions(sizes): time(1), x(1400), y(880), reference_time(1)
    variables(dimensions): int32 time(time), int32 reference_
↪time(reference_time), float64 x(x), float64 y(y), |S1 crs(), int32␣
↪index(y, x), float32 streamflow(time, y, x)
    groups:
```

6.5.1 LRGV Map-Grid Output

The LRGV, TX `CHRTOUT_GRID` *NetCDF* datafile was then read using the *Clover* `map_grid` module and displayed in the figure below. Stream thickness is determined by streamflow magnitude. The splotches covering the Laguna Madre are indicative of significant outliers being produced as artifcats of the LRGV domain.

```
/home/anernest/Git/DataExtractor/.venv/lib/python3.8/site-packages/
↪matplotlib/collections.py:981: RuntimeWarning: invalid value␣
↪encountered in sqrt
  scale = np.sqrt(self._sizes) * dpi / 72.0 * self._factor
Grid File 202111011300.CHRTOUT_GRID1
```

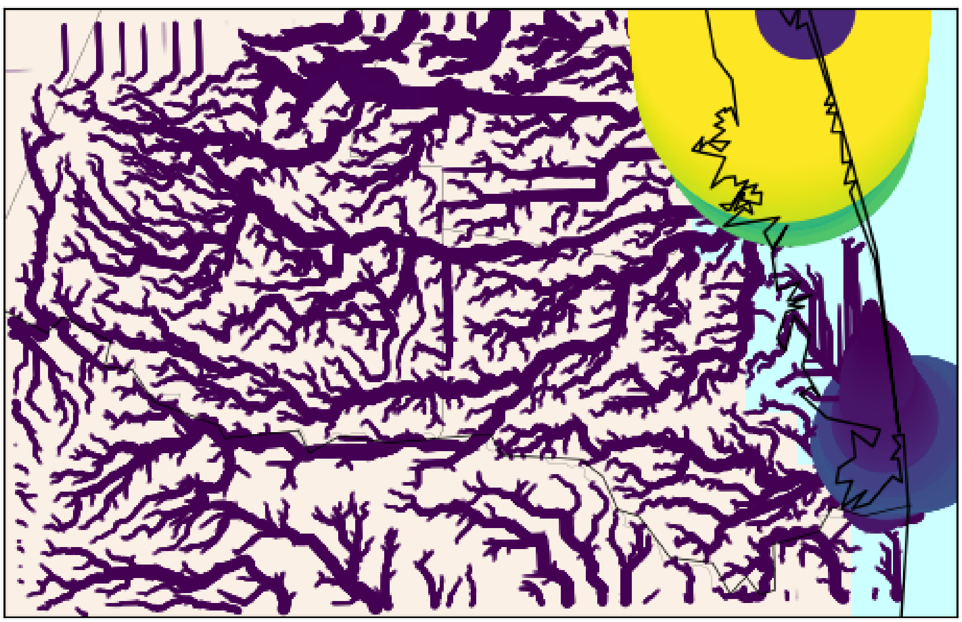


Figure6.6: LRGV Stream Network

6.5.2 LRGV Outlier Analysis

An outlier analysis was conducted on the same dataset. The table below depicts the output of the `nc_outlier_analysis` module. Filtering with a reasonable `m` does not appear to improve the dataset, indcating that the distribution is dominated by outliers.

..	min	max	mean	stdev
0	0	12656.1	59.228	629.748
1	0	688.715	7.05464	37.8911
2	0	1318.07	11.7668	77.1795
3	0	1942.36	14.3826	100.29
4	0	2573.13	16.3079	119.551

6.5.3 LRGV Streamflow Distribution (m=0)

The unfiltered streamflow is plotted in the chart below, showing a variability in predicted streamflow of 4 orders of magnitde. The miniscule number of streamflow records at the higher orders of magnitude indicate these to be model artifcats rather than representative of the physical system.

6.5.4 LRGV Streamflow Distribution (m=2)

The streamflow, filtered within 2 standard deviations, is plotted in the chart below, reducing the range by a single order of magnitde. The inadequacy of the filtering method indicates that a more directed filtering or masking technique must be employed to ensure that the resulting data is reflective of the physical system and imposing them as boundary conditions in *HEC-RAS* will positively affect the accuracy of the hydraulic model output.

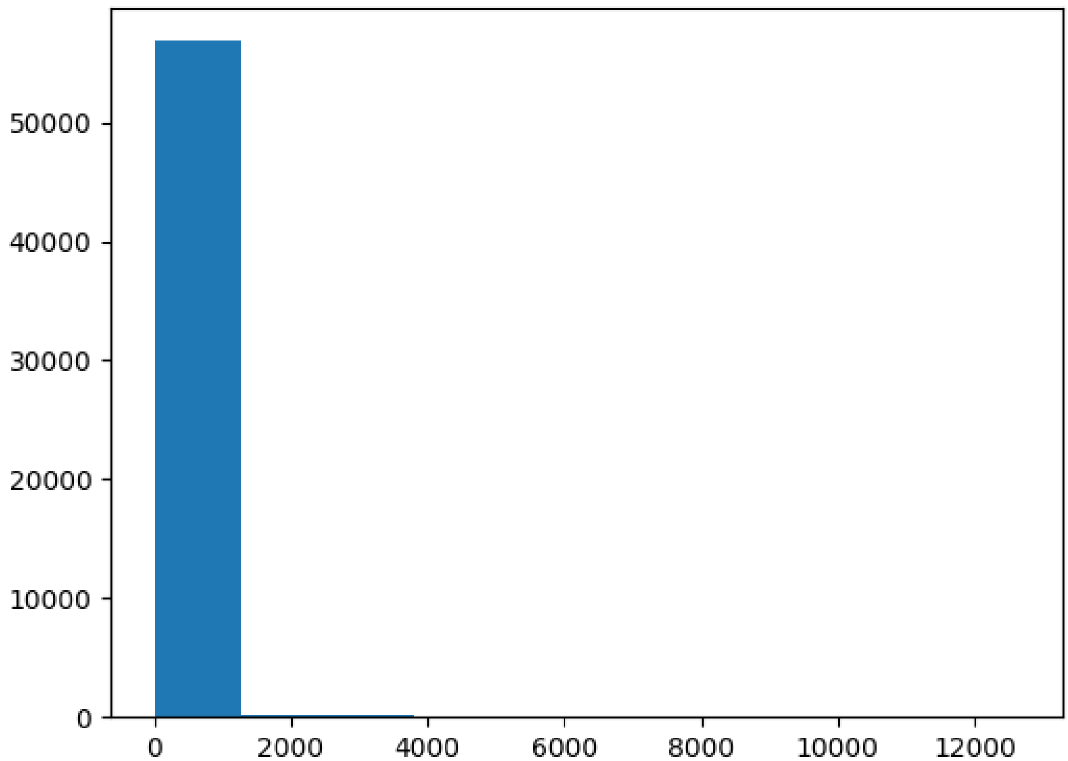


Figure6.7: LRGV, TX Unfiltered Streamflow Distribution

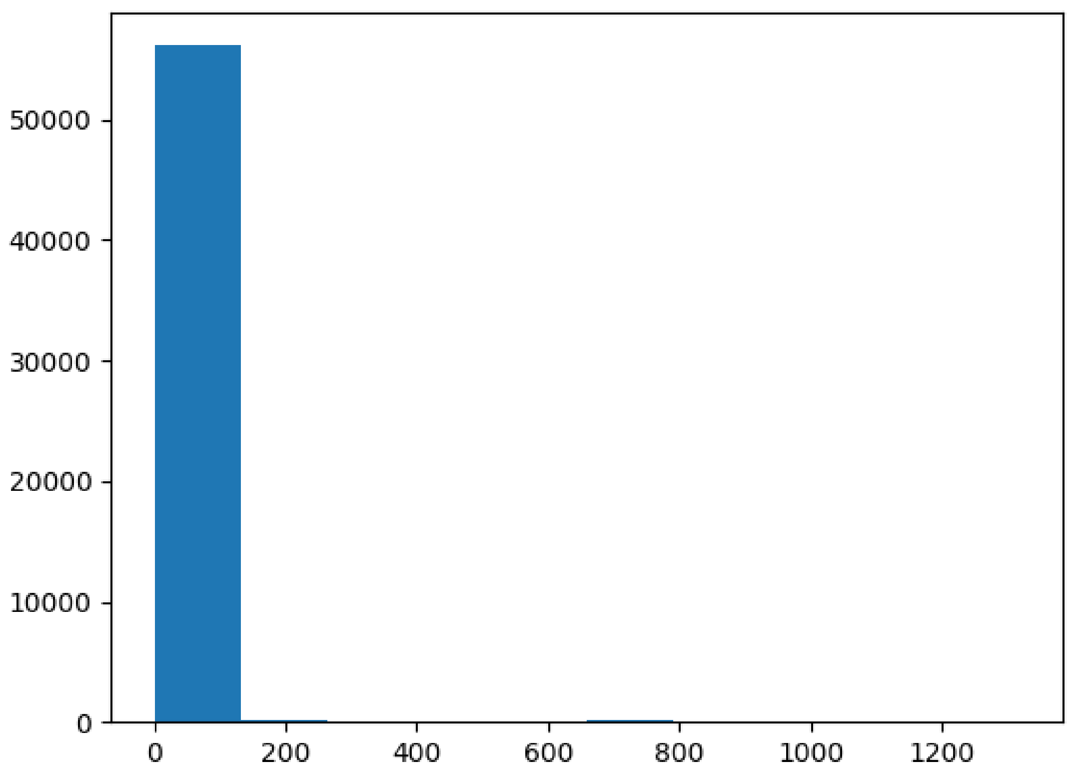


Figure6.8: LRGV, TX Streamflow Distribution Filtered within 2 Standard Deviations

6.5.5 LRGV Hydrograph

The LRGV, TX data was then plotted against time to produce the hydrograph below. Gridded output from *WRF-Hydro* was generated only in 2015 and 2021, resulting in the depicted data gap.

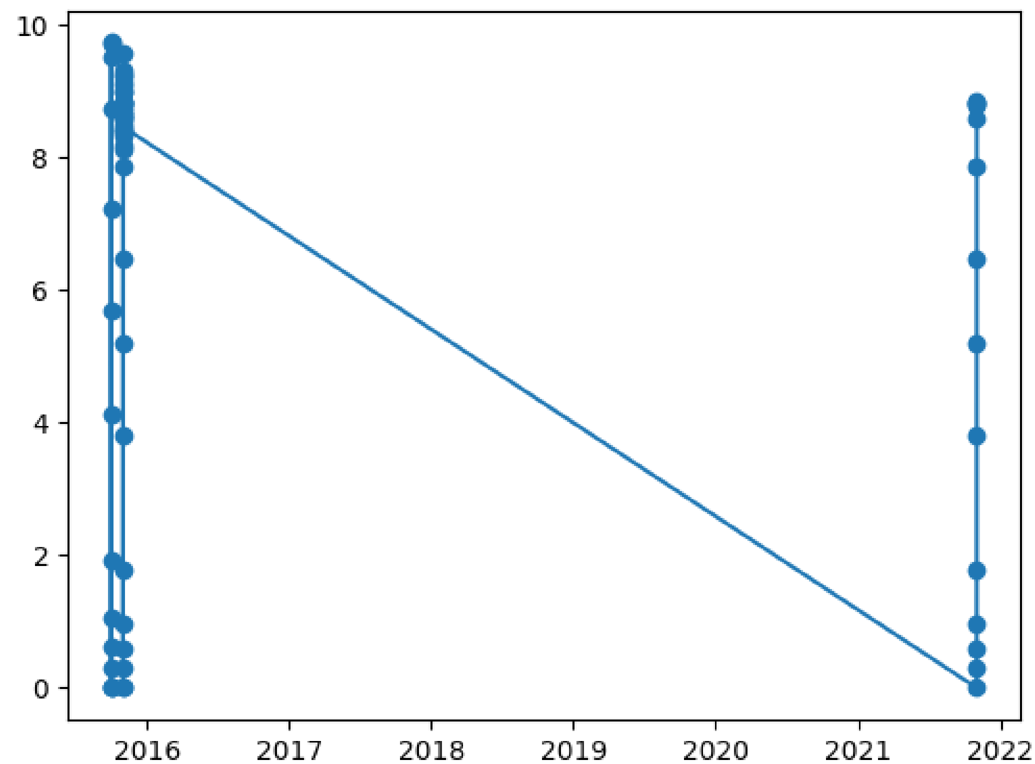


Figure6.9: LRGV Hydrograph

CHAPTER

SEVEN

CONCLUSIONS

1. A review of *WRF-Hydro* post-processing tools indicate that automated data extraction and ingestion will require development of new modules. These modules are being incorporated into *Clover* and have been successfully demonstrated for data analysis.
2. An assessment of the *HEC-DSS* file format indicates that while the format specifications are not readily available, the existence of a *FOSS* toolset indicates that the creation of *HEC-DSS* for use as *HEC-RAS* boundary condition files is a possibility.
3. An analysis of initial *LRGV WRF-Hydro* output indicates that the presence of model artifacts and outliers are likely to prove problematic if incorporated into *HEC-RAS* boundary condition files unfiltered or unmasked.

APPENDIX

A

GLOSSARY

API
Application Programming Interface

API.RGVFlood.com
RGVFlood.com data assimilation service.

AWS
Amazon Web Services

Azure
Microsoft's Cloud Computing Platform

Bernoulli
The Bernoulli equation is a simplification of the Navier-Stokes equations assuming inviscid fluid and steady (non-time-variant) flow.

BLE
Base Level Engineering

Celery
A task scheduling and messaging application used to maximize parallel task processing.

CentOS
A *Linux* distribution

CI
Cyberinfrastructure

Clover
Cloud Virtual Water Model Executor

COP
Common Operating Picture

CPU
Centralized Processing Unit

Crowdsource
Data collection from open, relatively un-controlled, sources.

CUAHSI
Consortium of Universities for the Advancement of Hydrologic Science

Cyberinfrastructure
computing systems, data storage systems, advanced instruments and data repositories, visualization environments, and people, all linked by high speed networks

DEM
Digital Elevation Model

Deterministic
Approaches to describing processes that do not rely on randomness.

DFIRM
Digital Flood Insurance Rate Map

DHS
Department of Homeland Security

DIKW
Data, Information, Knowledge, Wisdom

Django
<https://www.djangoproject.com/>

Docker
Docker is a container deployment platform that allows for the rapid deployment of a applications in the cloud, independent of the physical infrastructure.

DRF
Django ReST Framework

DSS
Decision Support System

EC2
AWS Elastic Cloud Compute

Eeyore
URL: Eeyore.ratesresearch.org CPU: Dual Intel(R) Xeon(R) E-2124 CPU @ 3.30GHz Memory: 16GB HD: 4TB OS: Ubuntu Linux 20.04

FEMA
Federal Emergency Management Agency

FIF
Flood Infrastructure Fund

FOSS
Free and Open Source Software

GCE
Google Compute Engine

GCP
Google Cloud Platform

GCS
Google Cloud storage

GeoNode
<https://geonode.org/>

GeoNode/db
PostgreSQL with *PostGIS* extensions database server storing *GeoNode Django* and *GeoServer* data.

GeoServer
Open source server for sharing geospatial data.

GeoTIFF
A public domain metadata standard which has the georeferencing information embedded within the *TIFF* file.

GIS
Geospatial Information System

GKE
Google *Kubernetes* Engine

H&H
Hydrologic and Hydraulic

HAND
Height Above Nearest Drainage <http://handmodel.ccst.inpe.br/>

HEC
Hydrologic Engineering Center

HEC-DSS
HEC Data Storage System

HEC-HMS
Hydrologic Engineering Center Hydrologic Modeling System. <https://www.hec.usace.army.mil/software/hec-hms/>

HEC-RAS
Hydrologic Engineering Center River Analysis System. <https://www.hec.usace.army.mil/software/hec-ras/>

HEC-RTS
Hydrologic Engineering Center Real Time Simulation

HPC
High Performace Computing

HPCC
: *HPC* cluster

HTML
: Hypertext Markup Language

HUC
: Hydrologic Unit Code

IDV
: Integrated Data Viewer from *UniData*

InfoWorks ICM
: <https://www.innovyze.com/en-us/products/infoworks-icm>

IT
: Information Technology

K8s
: *Kubernetes*

Kubernetes
: An orchestration system facilites the deployment and management of containerized applications, with a specific focus on scaling to increase demand for the provided services.

LaTeX
: A high-quality typesetting system including features designed for the production of technical and scientific documentation

LiDAR
: Light Detection and Ranging

Linux
: An open source operating system that is made up of the kernel, the base component of the OS, and the tools, apps, and services bundled along with it.

LLM/BSC
: Lower Laguna Madre/Brownsville Ship Channel watershed.

LRGV
: Lower Rio Grande Valley

LRGVDC
: Lower Rio Grande Valley Development Council

LSM
: Land Surface Models focus on describing the processes driving the exchange of terrestrial water with atmospheric.

Mechanistic
: Formulations describing physical, biological or chemical processes based on a theoretical understanding.

MIKE Urban+
<https://www.mikepoweredbydhi.com/download/mike-2019/mike-urban-plus?ref=%7B5399F5D6-40C6-4BB2-8311-37B615A652C6%7D>

MPI
Message Passing Interface

NAT
Network Address Translation

Navier-Stokes
The Navier-Stokes equations are mathematically representations of conservation of mass and momentum for simple fluids such as water.

NCAR
National Center for Atmospheric Research

NetCDF
NetCDF (Network Common Data Form) is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. It is also a community standard for sharing scientific data. The Unidata Program Center supports and maintains netCDF programming interfaces for C, C++, Java, and Fortran. Programming interfaces are also available for Python, IDL, MATLAB, R, Ruby, and Perl. Reproduced from **`NetCDF`_**.

NGINX
High performance web server.

NIC
Network interface controller

NLDAS
North American Land Data Assimilation System

NOAA
National Oceanic and Atmospheric Agency

NWC
National Water Center

NWM
National Water Model

NWS
National Weather Service

ODM
Observations Data Model

PostGIS
Spatial database extender for *PostgreSQL*

PostgreSQL
Open source object-relational database system, available with *PostGIS* extensions

Primo
Parallel raster inundation model

PWA
Progressive Web Application, an application format that allows installation as native applications onto mobile devices and desktop PCs directly from the web.

Python
<https://www.python.org/>

R
A language and environment for statistical computing and graphics

R
A language and environment for statistical computing and graphics.

RabbitMQ
An open-source inter-process message broker

RATES
Research, Applied Technology, Education and Service, Inc., a non-profit technology-based company.

RBAC
Role Based Access Control

REON
River and Estuary Observation Network. A partnership of organizations, supported by cloud software, committed to furthering the Democratization of Water Intelligence by sharing water data, analytics and models for local and regional decision making.

REON.cc
Cloud-based cyber-infrastructure that supports *REON*'s goals.

REON/db
PostgreSQL with *PostGIS* extensions database server storing *REON* specific data for *RTHS*, *REON/WM* & *REON.cc* data.

REON/RGV
Instantiation of *REON* with specific application to the Lower Rio Grande Valley - this includes the collection of *RTHS* stations, the *REON* partners with a stake in the LRGV, and the application of the *REON/WM* to the *LRGV*.

REON/WM
REON Water Model

ReST
REpresentational State Transfer

RGVFlood
: Instantiation of the *REON* Cyberinfrastructure specific to the *LRGV*.

RGVFlood.com
: The domain name and *URL* for *RGVFlood*.

RTHS
: Real Time Hydrologic System

RTHS.us
: Cloud server of *RTHS* network data

RWRAC
: Regional Water Resources Advisory Committee

SA
: Situational Awareness

SaaS
: Software as a Service

SMT
: Simultaneous Multi-Threading

SONAR
: Sound Navigation Ranging, a technique for detecting and determining the distance and direction of underwater objects by acoustic means.

Sphinx
: Documentation generator supporting multiple output formats

SPRNT
: Simulation Program for River Networks

Spyce
: Smartphone Python Computing Environment

Stochastic
: Approaches to describing processes in statistical terms.

SWMM
: Stormwater Management Model

Tastypie
: a webservice *API* framework for *Django*

TGLO
: Texas General Land Office

Tier I
: Tier I Real-Time Regional Hydrologic Modeling Framework

Tier II
Tier II On-Demand Sub-Regional Hydraulic Modeling Framework

Tier III
Tier III Off-Line Urban Stormwater Modeling Framework

TIFF
Tag Image File Format, a computer file used to store raster graphics and image information.

Tigger
URL: Tigger.water-wizard.org CPU: Dual Intel(R) Xeon(R) CPU E3-1245 v3 @ 3.40GHz Memory: 16GB HD: 4TB OS: Ubuntu Linux 20.04

TIN
Triangular Irregular Networks are a form of vector-based digital geographic data and are constructed by triangulating a set of vertices.

TWDB
Texas Water Development Board

TWDB/FIF
The Texas Water Development Board Flood Infrastructure Fund.

Ubuntu
A *Linux* distribution

UCAR
University Corporation for Atmospheric Research

UI
User Interface

UniData
A *UCAR* community program focused on sharing geoscience data and the tools to access and visualize that data.

URL
Uniform Resource Locator

USACE
United States Army Corps of Engineers

USGS
United States Geological Survey

USIBWC
United States International Boundary Water Commission

vCPU
Virtual *CPU*

VIC
Variable Infiltration Capacity (VIC) Macroscale Hydrologic Model. <https://vic.readthedocs.io/en/master/>

VM
Virtual Machine

Water Wizard
A suite of decision support tools designed for regional decision makers.

Wizard.RGVFlood.com
A web, mobile and desktop client-side application that, working with the server-side components at *RGVFlood.com*, provides the end-user with the up-to-date analytics, visualization and decision support services from the core *REON.cc CI*.

WPS
WRF Preprocessing System

WRDA
Water Resources Development Act

WRF
Weather Research and Forecasting Model

WRF-Hydro
WRF Hydrological modeling system. <https://ral. .edu/projects/wrf_hydro/overview>

PYTHON MODULE INDEX

C

INDEX