Lesson 0 - Getting Started[¶](#gjdgxs)

Overview[¶](#30j0zll)

In this training, we aim to walk you through a hands-on calibration experiment with the Poconos watershed located in Pennsylvania. Specifically, we will be calibrating the basin using observed and modeled streamflow at a USGS location over a few months in 2011 and validate over the remaining period in 2011. The short time period is for demonstration purposes only and not the common practice of the NWM calibration. Calibration of the NWM is done using a 5-year period (2008-10 to 2013-10) in order to cover different climate regimes in the calibration period. Details of domains, and calibration experiments for this exercise will be provided in this and the upcoming lessons.

The calibration procedure relies on a python library named PyWrfHydroCalib, which explicitly handles the setup, model tuning, and provides diagnostics supporting the calibration process. This lesson is intended to provide an overview of few key definitions, system requirements, and orient you through the calibration material provided. We will describe the calibration procedure and fundamentals in Lesson 1, and follow up with hands-on experiment in Lesson 3 and 4 where we will setup, calibrate and validate Pocono watershed.

Key Terms[¶](#1fob9te)

### WRF-Hydro:[¶](#3znysh7)

An open-source community model, used for a range of projects, including flash flood prediction, regional hydroclimate impacts assessment, seasonal forecasting of water resources, and land-atmosphere coupling studies.

### The National Water Model (NWM):[¶](#2et92p0)

A joint development effort between the National Center for Atmospheric Research (NCAR) and the National Weather Service (NWS) Office of Water Prediction (OWP). The NWM is a set of operational analysis and forecast configurations of the WRF-Hydro modeling system (Gochis et al., 2013) which runs over the entire continental United States (<https://water.noaa.gov/about/nwm>) as well as Hawaii and Puerto Rico. Alaska will also become operational in NWMv30.

### PyWrfHydroCalib:[¶](#tyjcwt)

A robust workflow for calibrating parameters over individual basins with a focus on automating the process for hundreds of forecast points in the NWM domain. Here, we describe the various steps involved in setting up a WRF-Hydro calibration experiment, along with how to execute the calibration workflow and access the various results from it. This version of the calibration workflow is the RnD branch of the NWMv30 calibration and it not yet finalized and official. There are still developments and code improvement underway and it will be finalized before our official NWMV30 onborading, scheduled currenlty for Jan 2022.

PyWrfHydroCalib Requirements[¶](#3dy6vkm)

Calibration goes beyond setting up the NWM or WRF-Hydro for a model simulation. Some understanding of the preparation of datasets, hydrologic parameters, and their associated impact on model states is needed. You will decide which parameters to calibrate, along with the proper ranges, and the length of the calibration simulation/evaluation period. It is highly encouraged that you run sensitivity analysis for the modeling domain to explore which parameters have significant impact on the modeled states being calibrated.

### Software requirements:[¶](#1t3h5sf)

#### Python Packages:[¶](#4d34og8)

To run the calibration workflow, there are multiple software requirements and pre-processing steps that need to take place prior to actual calibration. Currently, the calibration workflow relies on a set of Python and R libraries to execute various functions and plotting capabilities. This is in addition to existing requirements to compile and run WRF-Hydro/NWM. As of early 2019, the calibration workflow code has been migrated from Python 2 to Python 3. The following Python libraries are used by the workflow:

1.) numpy 2.) NetCDF4 (for Python) 3.) pandas 4.) psutil 5.) psycopg2 6.) xarray

In [1]:

*# Optional: Verify Python Libraries*  
**import** numpy  
**import** netCDF4  
**import** pandas  
**import** psutil  
**import** psycopg2  
**import** xarray  
print('all python packages accounted for')

all python packages accounted for

R Libraries[¶](#2s8eyo1)

For R, there are several additional libraries required for proper analysis and plotting to take place. The following R libraries are needed for proper execution:

1.) data.table 2.) ggplot2 3.) ncdf4 4.) plyr 5.) boot 6.) sensitivity 7.) randtoolbox 8.) gridExtra 9.) hydroGOF 10.) qmap 11.) zoo

12.) raster 13.) lubridate 14.) reshape2

In [ ]:

*# Verify R libraries (for Jupyter notebook on Cheyenne, switch to R kernel for now, will be updated for docker notebook)*  
library(data**.**table)  
library(ggplot2)  
library(ncdf4)  
library(plyr)  
library(gridExtra)  
library(hydroGOF)  
library(qmap)  
library(zoo)  
library(raster)  
library(lubridate)  
library(reshape2)  
library(boot) *# used for the sensitivity analysis*   
library(sensitivity) *# used for the sensitivity analysis*   
library(randtoolbox) *# used for the sensitivity analysis*

Prerequisites:[¶](#17dp8vu)

1. Installation of the WRF-Hydro/NWM modeling system (use a version that is indicated in the release notes of the package) See the 'WRF-Hydro Technical Description and User Guide' available from the WRF-Hydro Modeling System website for details: <https://ral.ucar.edu/projects/wrf_hydro>.
2. Preparation of a modeling domain.
3. Preparation of associated forcing files necessary to run the model simulations.
4. Preparation of input observation files that will be used in the calibration workflow.

*Note: All of these files are provided for you in this training. We have provided the WRF-Hydro code that has already compiled. There is also a supplemental lesson (Lesson-S1-wrf\_hydro\_compile.ipynb) that walk you through the model compilation if you wish to do so.*

### Which files will be updated during calibration?[¶](#3rdcrjn)

During the calibration process we will be updating the parameters that exist in one of the following files.

* *soil\_properties.nc* : A 2D property file that contains values impacting sub-surface hydrologic response.
* *HYDRO\_TBL\_2D.nc* : The 2D NetCDF of surface hydrologic parameters impacting hydrologic response for overland flow routing.
* *Fulldom.nc* : The 2D geospatial fabric utilized for the high-resolution routing.
* *GWBUCKPARM.nc* : Groundwater bucket parameter file.

These files are located in the domain directory that are prepared for you, and will be discussed in detail later.

## Orientation to training material[¶](#26in1rg)

All the required codes and domain files are being prepared for you in advance. Let's take a look at the content of the materials.

In [2]:

%%bash  
ls /home/docker

example\_case  
lessons  
miniconda3  
PyWrfHydroCalib  
WrfHydroForcing  
wrf\_hydro\_nwm\_public  
wrf\_hydro\_training

**lessons**: This directory contains all the lessons that we will use in this training as well as several supplementary lessons that you could use on your own to explore more.

**wrf\_hydro\_nwm\_public**: This directory contains the WRF\_Hydro code that is already compiled for you. Refer to supplemental lesson (Lesson-S1-wrf\_hydro\_compile.ipynb) for details on how to compile the model. Let's take a look at the content of the Run folder that is created after model compilations and contains the model executable as well as namelists and table files that are compatible with the compiled version of the code. *Note* that we will use the TBL files from this Run directory, however, the namelists will be generated by the python workflow (PyWrfHydroCalib).

In [3]:

%%bash  
ls /home/docker/wrf\_hydro\_nwm\_public/trunk/NDHMS/Run

CHANPARM.TBL  
GENPARM.TBL  
hydro.namelist  
HYDRO.TBL  
MPTABLE.TBL  
namelist.hrldas  
SOILPARM.TBL  
wrf\_hydro.exe  
wrf\_hydro\_NoahMP.exe

**PyWrfHydroCalib**: This directory contains the python package that is used in model calibration. The main scripts (called for workflow managements) are placed in the top level directory, and core directory has all the functions and R scripts that are called by the top level python scripts.

*Note: there is a manual placed in the top level directory, however, this manual has not been updated with the NWMv30 developments and is out of date. There has been a number of code changes that is not yet documented.*

In [4]:

%%bash  
ls /home/docker/PyWrfHydroCalib  
  
echo -------- Setup\_files --------------  
  
ls /home/docker/PyWrfHydroCalib/setup\_files/

calibOrchestrator.py  
calib.py  
core  
initDB.py  
inputDomainMeta.py  
jobInit.py  
LICENSE.md  
PyWrfHydroCalib\_User\_Guide.pdf  
README.md  
runValidOrchestrator.py  
sensitivity.py  
setup\_files  
spinOrchestrator.py  
spinup.py  
util  
validation.py  
validOrchestrator.py  
-------- Setup\_files --------------  
calib\_params.tbl  
domainMeta\_01447680.csv  
domainMeta\_01447720.csv  
gage\_list\_template.csv  
sens\_params.tbl  
setup.parm

**what is new in PyWrfHydroCalib v30?**:

* multi site calibration capability
* step wise calibration of the nested gages
* snow calibration (combination of streamflow and snow used in the objective function)
* soil moisture calibration (combination of streamflow and soil moisture error metrics used in the objective function)
* newly added metrics such categorical metrics and event based metrics
* bug fixes and improvement of the procedure

**exmaple\_case**: This lesson will use a prepared domain located in the ~/example\_case directory. The structure of the example\_case directory serves as a good example of how to organize your input files for calibration. If using another domain with this lesson, such as another NWM cutout, it is imperative that the file names and directory structure match those described below. Under the example\_case there are two domains with the default parameters that are cutouts of NWM CONUS domains. Figure below show the location of these basins. Note that there are two lakes in the domain on the 01447720 branch. 01447720 has been calibrated in the previous versions of NWM and will be used for our training calibration exercise.



Let's take a look at the content of the domain directory.

In [5]:

%%bash  
ls /home/docker/example\_case/Calibration/Input\_Files/01447720/

FORCING  
Fulldom.nc  
geo\_em.nc  
GEOGRID\_LDASOUT\_Spatial\_Metadata.nc  
GWBUCKPARM.nc  
HYDRO\_TBL\_2D.nc  
LAKEPARM.nc  
OBS  
params.txt  
RouteLink.nc  
script\_forcing\_subset.txt  
script\_forcing\_subset.txt\_NWM\_REALTIME  
soil\_properties.nc  
spatialweights.nc  
wrfinput.nc

**FORCING**: This directory contains all the forcing data for our simulation.

**Domain Files**: The table below summarize the domain files that are required for model run and also identifies the ones that get updated during calibration process.

| **Filename** | **Description** | **Source** | **Will be updated in calibration?** |
| --- | --- | --- | --- |
| geo\_em.nc | Data required to define the domain and geospatial attributes of a spatially-distributed, or gridded, 1-dimensional (vertical) land surface model (LSM) | GEOGRID utility in the WRF preprocessing system (WPS) | No |
| wrfinput.nc | file including all necessary fields for the Noah-MP land surface model, but with spatially uniform initial conditions. Users should be aware that the model will likely require additional spin-up time when initialized from this file. | create\_Wrfinput.R script | No |
| Fulldom.nc | High resolution full domain file. Includes all fields specified on the routing grid. | WRF-Hydro GIS pre-processing toolkit with some custom modification | Yes |
| RouteLink.nc | This file contains all the information and required parameters of reaches required for channel routing | based on the NHDPlus and other custom hydrography dataets | No |
| spatialweights.nc | netCDF file specifying the weights to map between the land surface grid and the pre-defined groundwater basin boundaries | custom python script | No |
| LAKEPARM.nc | Lake parameter table containing lake model parameters for each catchment | WRF-Hydro GIS pre-processing toolkit | No |
| GWBUCKPARM.nc | Groundwater parameter table containing bucket model parameters for each basin | WRF-Hydro GIS pre-processing toolkit | Yes |
| hydro2dtbl.nc | Spatially distributed parameter table for lateral flow routing within WRF-Hydro. | create\_SoilProperties.R script (will also be automatically generated by WRF-Hydro) | Yes |
| soil\_properties.nc | Spatially distributed land surface model parameters | create\_SoilProperties.R script | Yes |
| GEOGRID\_LDASOUT\_Spatial\_Metadata.nc | projection and coordinate information for the land surface model grid. | WRF-Hydro GIS pre-processing toolkit | No |

Both FORCING and DOMAIN directories are created using the subsetting scripts developed by NCAR.

**Note**: The rest of the files in the DOMAIN directory that are not explained in the table above are created by the subsetting scripts when subsetting this domain from the NWM CONUS domain and are not used by the model or calibration workflow.

**OBS**: This directory contains all the observation data for our experiment. These are Rdatasets that are ready by the R scripts.

* obsStrData.Rdata: observation file for streamflow.
* obsSnowData.Rdata: observation file for mean areal SWE.
* obsSoilData.Rdata: observation file for mean areal soil moisture.

It is the user's responsibility to prepare the files and they need to comply with the format expected. Let us look at the content of these files, starting with the streamflow file:

| **agency\_cd** | **site\_no** | **POSIXct** | **obs** | **quality\_flag** | **threshold** |
| --- | --- | --- | --- | --- | --- |
| USGS | 01447720 | 2007-10-01 04:00:00 | 2.010496 | NA | 23.3758 |
| USGS | 01447720 | 2007-10-01 05:00:00 | 1.999169 | NA | 23.3758 |
| USGS | 01447720 | 2007-10-01 06:00:00 | 1.999169 | NA | 23.3758 |
| USGS | 01447720 | 2007-10-01 07:00:00 | 1.999169 | NA | 23.3758 |
| USGS | 01447720 | 2007-10-01 08:00:00 | 1.999169 | NA | 23.3758 |
| USGS | 01447720 | 2007-10-01 09:00:00 | 1.999169 | NA | 23.3758 |

Below is a short description of the content:

* *agency\_cd* : Not required.
* *site\_no* : USGS gage identifier.
* *POSIXct* : Time of the observation, the class should be POSIXct as the name suggest.
* *obs* : Streamflow observed values in cms.
* *quality\_flag* : Not required.
* *threshold* : Used in the calculation of categorical metrics POD, FAR and CSI and we have used the 95% as the threshold.

Next, let us look at the obsSnowData.Rdata content, table below has the first few lines of the data. Like obsStrData.Rdata, the quality flag and agency\_cd are not required. the obs in this table is the mean areal SWE values from SNODAS product that are reported at 6 AM and will be paired with the model simulations at that time if the user turns on the snow calibration.

| **site\_no** | **obs** | **POSIXct** | **agency\_cd** | **quality\_flag** |
| --- | --- | --- | --- | --- |
| 01447720 | 0 | 2004-09-01 06:00:00 | SNODAS | NA |
| 01447720 | 0 | 2004-09-02 06:00:00 | SNODAS | NA |
| 01447720 | 0 | 2004-09-03 06:00:00 | SNODAS | NA |
| 01447720 | 0 | 2004-09-04 06:00:00 | SNODAS | NA |
| 01447720 | 0 | 2004-09-05 06:00:00 | SNODAS | NA |
| 01447720 | 0 | 2004-09-06 06:00:00 | SNODAS | NA |

Next, we will look at the obsSoilData.Rdata cotent, table below has the first few lines of the data. Like obsStrData.Rdata, the quality flag and agency\_cd are not required. the obs in this table is the mean areal soil moisture observed by SMAP averaged over the day. There are few extra steps taken before pairing the model simulated soil moisture at the top layer (10 cm) and SMAP soil moisture (top 5 cm), such as sliding window averaging and CDF matching. We will discuss the details in a different lecture.

| **site\_no** | **Date** | **obs** | **agency\_cd** | **quality\_flag** |
| --- | --- | --- | --- | --- |
| 01447720 | 2015-04-01 | NA | SMAP | NA |
| 01447720 | 2015-04-02 | NA | SMAP | NA |
| 01447720 | 2015-04-03 | 0.3494052 | SMAP | NA |
| 01447720 | 2015-04-04 | 0.3844350 | SMAP | NA |
| 01447720 | 2015-04-05 | NA | SMAP | NA |
| 01447720 | 2015-04-06 | 0.4300236 | SMAP | NA |

**WARNING**: name of the files and structure of the domain directory are hardcoded in the calibration python workflow and the user is expected to follow the convention used above, otherwise the calibration workflow will not be able to find the files and will result in an error.

Conclusion:[¶](#lnxbz9)

We reviewed the inputs to the calibration workflow that were prepared for use in this training. Proceed to *Lesson 1 - Calibration Overview*

In [ ]: