**Empirical Local Patch**

1. Make a directory called **$(CaMa-Flood-DA).**
2. You can copy the **$(CaMa-Flood-DA)** directory directly from the rainbow server. [Location: /cluster/data8/abdul.moiz/20230601\_CaMa-DA\_shared]. This is preferred since this repository has all the updated codes. Alternatively you can follow steps 3-5 to download the repository from GitHub.
3. Set up the CaMa-Flood model in **$(CaMa-Flood-DA)/$(CaMa-Flood)**.
4. Create a new directory **$(CaMa-Flood-DA)/$(Empirical-Local-Patch)**.
5. Go to **$(CaMa-Flood-DA)/$(Empirical-Local-Patch)** and download the GitHub repository by running ‘git clone <https://github.com/MenakaRevel/Empirical_LocalPatch.git>’
6. Make shell scripts executable by running ‘chmod +x \*.sh’
7. Modify params.py for parameters for your Empirical Local Patch setup. For the example of Amazon simulation, the following parameters were used:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Value** |
| starttime | Starting date for CaMa-Flood simulations | [2000,1,1] |
| endtime | Ending date for CaMa-Flood simulations | [2020,12,31] |
| CaMa\_dir | CaMa-Flood model directory | $(CaMa-Flood-DA)/$(CaMa-Flood) |
| out\_dir | Empirical Local Patch directory | $(CaMa-Flood-DA)/$(Empirical-Local-Patch) |
| map\_name | Name of the CaMa-Flood map used | ‘amz\_06min’ |
| input\_name | Name of runoff input used | ‘ERA5’ |
| spinup\_mode | Switch for spinup | 1 (for doing spinup simulation)  0 (for no doing spinup simulation) |
| spinup\_end\_year | End year for spinup | 2000 |
| spinup\_end\_month | End month for spinup | 12 |
| spinup\_end\_date | End date for spinup | 31 |
| patch\_start | Start date for empirical local path | 2000,1,1 |
| patch\_end | End date for empirical local path | 2020,12,31 |
| threshold | Weight threshold for empirical local patch | 0.6 |
| cpu\_nums | Number of CPUs used for calculation | 40 |
| damrep | Switch for dam representation | 1 (represent dams in local patch)  0 (do not represent dams in local patch) |

1. Link input runoff data by running ‘**ln -s $(CaMa-Flood-DA)/$(CaMa-Flood)/inp/ $(CaMa-Flood-DA)/$(Empirical-Local-Patch)/CaMa\_in**’
2. Modify **s01-CaMa\_sim.sh** for **$(DROFUNIT)**, **$(CROFDIR)**, **$(CROFPRE)**, **$(CROFSUF)**, **$(CDIMINFO)** and **$(CINPMAT)** following CaMa-Flood documentation. For this example, the following values were used:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Descriptions** | **Values** |
| DROFUNIT | See CaMa-Flood documentation | "86400" |
| CROFDIR | See CaMa-Flood documentation | "${INBASE}/CaMa\_in/"${inputname} |
| CROFPRE | See CaMa-Flood documentation | "Roff\_\_\_\_" |
| CROFSUF | See CaMa-Flood documentation | ".sixmin" |
| CDIMINFO | See CaMa-Flood documentation | "${FMAP}/diminfo\_amz-06min.txt" |
| CINPMAT | See CaMa-Flood documentation | ${FMAP}/inpmat\_amz-06min.bin |

1. Before proceeding with the following steps, make sure to modify the PBS settings in each script if you intend to use qsub.
2. Execute **s01-CaMa\_sim.sh** by running **‘./s01-CaMa\_sim.sh’.** This script simply runs the CaMa-Flood model for the specified period.
3. Go to **src/** and compile all the code. Run **‘make clean’** and then **‘make all’.**
4. Execute **s02-conv\_bin2nc.sh** by running **‘./s02-conv\_bin2nc.sh’.** This script converts the binary output from CaMa-Flood to netCDF.
5. Execute **s03-remove\_trend.sh** by running **‘./s03-remove\_trend.sh’.** This script generates the trend removed CaMa-Flood output. For example, **‘rmdtrnd2000-2020.nc’**.
6. Execute **s04-remove\_season.sh** by running **‘./s04-remove\_season.sh’.** This script removes seasonality from **‘rmdtrnd2000-2020.nc’** and generates **‘rmdsesn2000-2020.nc’.**
7. Execute **s05-standardize.sh** by running **‘./s05-standardize.sh’.** This script standardizes **‘rmdsesn2000-2020.nc’** and generates **‘standardized2000-2020.nc’.**
8. Execute **s06-semivariance.sh** by running **‘./s06-semivariance.sh’.** This script calculates the semivariogram and generates a directory for each grid of CaMa-Flood inside the **$(CaMa-Flood-DA)/$(Empirical-Local-Patch)/semivar** directory.
9. Go to **/etc/src** and compile all the code by running **‘make clean’** and **‘make all’**
10. Link GRanD database by running **‘ln -s /home/yamadai/work/data/Dam+Lake/GRanD/inp/GRanD\_v1\_1\_inp.csv dam\_inp/damlist.csv’**
11. Execute **s07-weightage.sh** by running **‘./s07-weightage.sh’.** This will generate spatial weights and gaussian weights in the **$(CaMa-Flood-DA)/$(Empirical-Local-Patch)/weightage** and **$(CaMa-Flood-DA)/$(Empirical-Local-Patch)/gaussian\_weight** directory, respectively.
12. Execute **s08-localiz\_para.sh** by running **‘./s08-localiz\_para.sh’.** This will generate empirical local patches in the **$(CaMa-Flood-DA)/$(Empirical-Local-Patch)/local\_patch.** In this directory there is a text file for each CaMa-Flood grid. Each text file has the CaMa-Flood coordinates and gaussian weights for the grids defining the local patch.
13. You can generate local patches only along the mainstream by executing **s09-localizMS\_para.sh.** [Note: This is not fully tested yet.]
14. Compress all the intermediate files by executing s10-archive.sh. Run **‘./s10-archive.sh’**. This step is not compulsory and takes quite a lot of time. However, you can save storage space by compressing a large number of files. This also allows for easier transfer of files, since a large number of files in compressed into one file.
15. You can plot a time series of the CaMa-Flood simulated water surface elevation, trend removed water surface elevation, seasonality removed water surface elevation and standardized water surface elevation by running **‘python p01\_plot\_timeseries.py’.** Plots are generated for each gauge location in the GRDC database in the region (e.g AMAZONAS). A sample output plot is shown in Figure 1.

A picture containing text, font, screenshot, typography

Description automatically generated

Figure 1 - Output of p01\_plot\_timeseries.py

1. To plot a semivariogram, run **‘python p02\_plot\_semivari.py’.** This plot a semivariogram for a river segment. A sample is shown in Figure 2.

A picture containing text, diagram, line, map

Description automatically generated

Figure 2 - Output of p02\_plot\_semivari.py

1. To plot a spatial map of the spatial weights and empirical local patch run **‘python p03\_plot\_local\_patch.py’.** This will generate a spatial weight map and empirical local patch map for various grids corresponding to the location of GRDC gauges. A sample output of spatial weight and local patch is shown in Figure 3a and Figure 3b respectively.

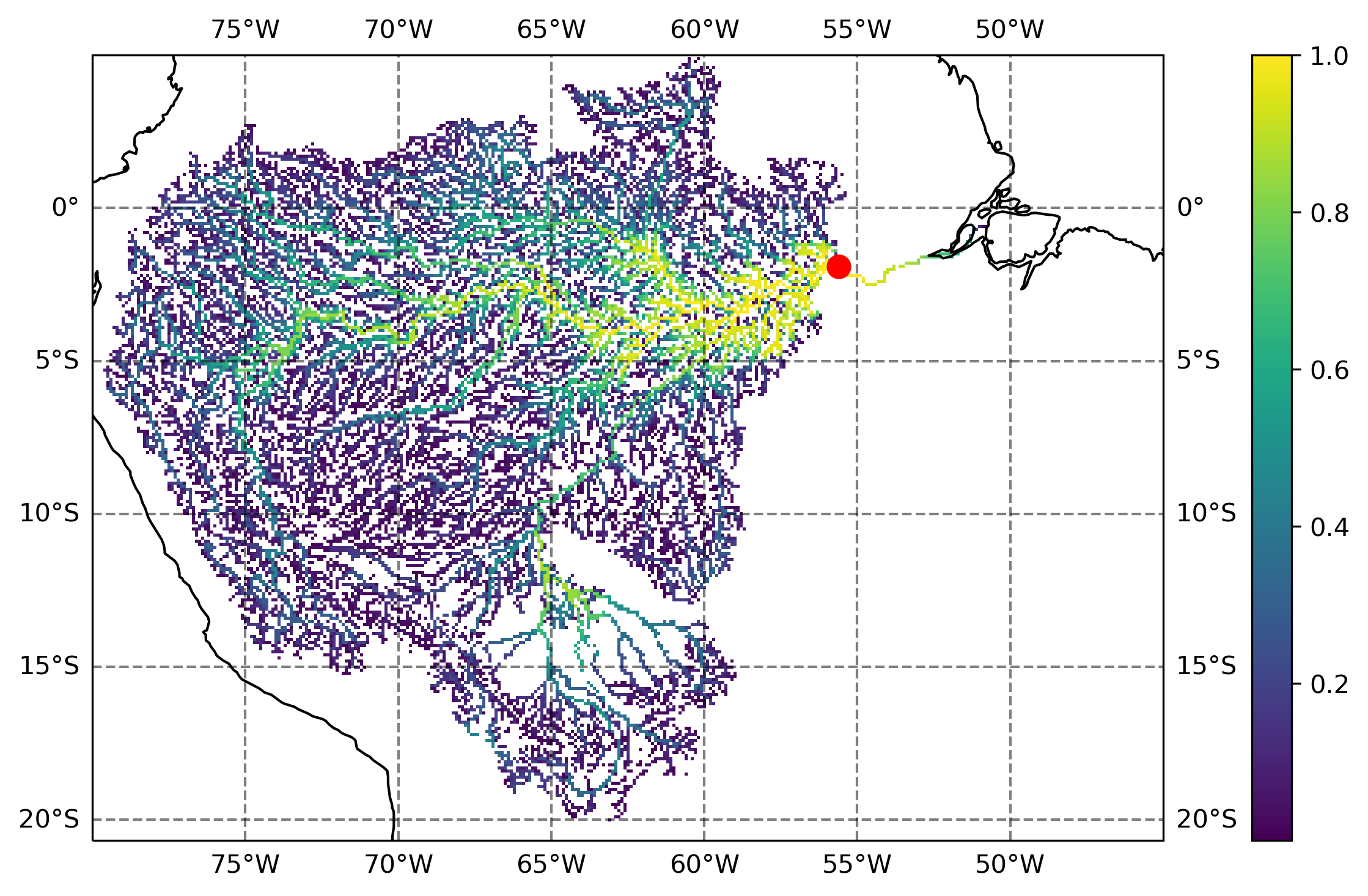
****

Figure 3a - Output of p03\_plot\_local\_patch.py showing spatial weights

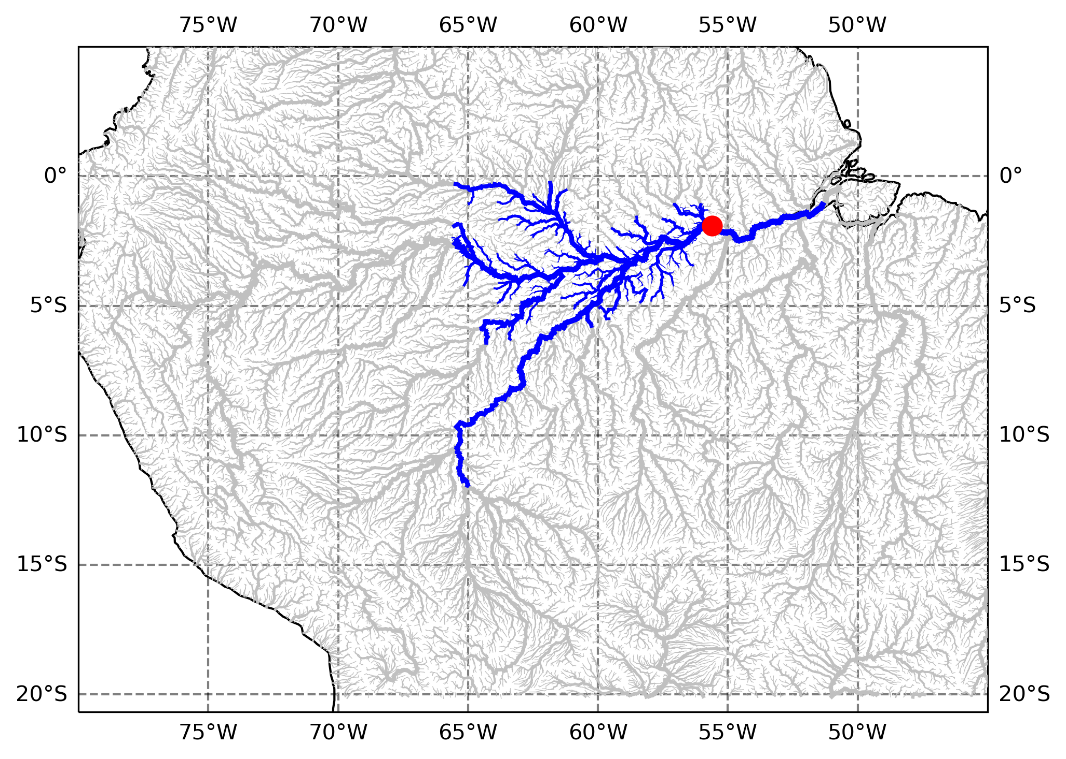


Figure 3b - Output of p03\_plot\_local\_patch.py showing local patch

**Ensemble Simulation**

1. Make a new directory **$(CaMa-Flood-DA)/$(Ensemble-Simulation)**.
2. Copy Ensemble Simulation code from the repository. [Location: /cluster/data8/abdul.moiz/20230601\_CaMa-DA\_shared]
3. Link Runoff data from CaMa-Flood directory. Run **‘ln -s $(CaMa-Flood-DA)/$(CaMa-Flood)/inp/ERA5 $(CaMa-Flood-DA)/ $(Ensemble-Simulation)/CaMa\_in/ERA5/bin**’
4. Go to **src/** and compile all the codes by running **‘make clean’** and **‘make all’**.
5. Modify **params.py** file. The following parameters are used for the sample Amazon simulation

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Values** |
| timestep | Number of seconds in a timestep | 86400 |
| starttime | Start date for ensemble simulation | (2000,1,1) |
| endtime | End date for ensemble simulation | (2020,12,31) |
| start\_year | Start year for ensemble simulation | 2000 |
| end\_year | End year for ensemble simulation | 2020 |
| org\_dir | Ensemble simulation directory | $(CaMa-Flood-DA)/ $(Ensemble-Simulation) |
| CaMa\_dir | CaMa-Flood model directory | $(CaMa-Flood-DA)/$(CaMa-Flood) |
| hydro\_da\_dir | HydroDA directory [For Data Assimilation step] | $(CaMa-Flood-DA)/$(Hydro-DA) |
| runname | Name of input runoff used | ‘ERA5’ |
| input | Name of input runoff used | ‘ERA5’ |
| map\_name | Name of CaMa-Flood map used | ‘amz\_06min’ |
| ens\_mem | Number of ensemble members | 20 |
| expname | Name of experiment | ‘AMZ’ |
| method | Method used to generate ensemble | ‘lognormal’ [‘simple’ and ‘normal’ are other options] |
| distopen | Corrupted runoff's percentage | 0.50 |
| diststd | Noise to make runoff input to scatter ensembles | 0.25 |
| beta | Parameter to generate ensembles | 0.0 |
| alpha | Parameter to generate ensembles | 0.993 |
| E | Error percentage | 0.30 |
| mode | Change depending upon input runoff | 4 |
| run\_flag |  | 0 |
| spinup\_flag |  | 0 |
| para\_nums |  | 6 |
| cpu\_nums | Number of CPUs to be used | 20 |

1. Make shell scripts executable by running ‘chmod +x \*.sh’
2. Execute s01-prep\_runoff.sh by running ‘./s01-prep\_runoff.sh’
3. Execute s02-run\_CaMa.sh by running ‘./s02-run\_CaMa.sh’
4. Execute s03-conv\_bin2nc.sh by running ‘./s03-conv\_bin2nc.sh’
5. Execute s04-mk\_stat.sh by running ‘./s04-mk\_stat.sh’
6. Execute s05-copy2HydroDA.sh by running ‘./s05-copy2HydroDA.sh’

**AltiMaP**

1. Make a new directory $(CaMa-Flood-DA)/$(AltiMaP).
2. Copy HydroWeb Directory. Include direct download of data.
3. Copy AltiMap [s0\* and a0\* scripts and src]
4. Do Chmod +x \*.sh to make all scripts executable.
5. Modify s01-prep\_VSlist.sh for the database that you want to use along with the location of that data directory (for example, HydroWeb). Execute s01-prep\_VSlist.sh
6. Link glb\_06min with full data maps (including 03sec maps) by doing doing ‘ln -s /work/a01/yamadai/work/CaMa\_v400/cmf\_v400\_data/map\_full/glb\_06min /cluster/data8/abdul.moiz/20230420\_CaMa-DA/CaMa-Flood\_v4/map/glb\_06min’
7. Execute s02-allocate\_VS.sh
8. Modify $obstxt in s03-unreal\_obs.sh with the appropriate name of the file (mainly need to change the date). Execute s03-unreal\_obs.sh

**HydroDA**

1. Make a new directory **$(CaMa-Flood-DA)/$(HydroDA)**. If you have already run the script **$(CaMa-Flood-DA)/$(Ensemble-Simulation)/s05-copy2HydroDA.sh**, then this directory should already be created.
2. Running **$(CaMa-Flood-DA)/$(Ensemble-Simulation)/s05-copy2HydroDA.sh** previously copies the mean\_sfcelv and std\_sfcelv in the **$(CaMa-Flood-DA)/$(HydroDA)/dat** directory.
3. Running **$(CaMa-Flood-DA)/$(AltiMaP)/copy2HydroDA\_org.py** copies the list of all the virtual stations in the target region to **$(CaMa-Flood-DA)/$(HydroDA)/dat/HydroWeb\_alloc\_amz\_06min\_org.txt**
4. Modify **$(CaMa-Flood-DA)/$(HydroDA)/etc/prep\_obs.py** with the following parameters for the case of this Amazon simulation:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Value** |
| starttime() | Start time for including satellite observations | 2014,1,1 |
| endtime() | Time upto which satellite observations will be included | 2015,12,31 |
| obs\_name() | Name of the satellite observation dataset | “HydroWeb” |
| CaMa\_dir() | Path to the CaMa-Flood model directory | $(CaMa-Flood-DA)/$(CaMa-Flood) |
| mapname() | Name of CaMa-Flood map used | “amz\_06min” |

1. Execute prep\_obs.py by running ‘python prep\_obs.py’. This creates a series of text files in **$(CaMa-Flood-DA)/$(HydroDA)/obs** for each day. Each text file has the ix, iy, value of satellite observation, mean, standard deviation and the name of the satellite observation respectively.
2. Go to **$(CaMa-Flood-DA)/$(HydroDA)/gosh** and compile all the codes for HydroDA by running ‘sh compile.sh yes’
3. Modify the parameters for HydroDA **$(CaMa-Flood-DA)/$(HydroDA)/gosh/params\_real.py** with the following values for the case of the Amazon simulation:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Value** |
| mode() | changes configuration depending upon the runoff dataset used | 5 (for ERA5) |
| conflag() | Flag for observation conversion | 3 (for Normalized Values) |
| mapname() | Name of CaMa-Flood map used | “amz\_06min” |
| CaMa\_dir() | Path to the CaMa-Flood model directory | $(CaMa-Flood-DA)/$(CaMa-Flood) |
| assimS() | South bound for the region to be assimilated | -20 (degrees) |
| assimN() | North bound for the region to be assimilated | 5 (degrees) |
| assimW() | West bound for the region to be assimilated | -80 (degrees) |
| assimE() | East bound for the region to be assimilated | -45 (degrees) |
| patch\_size() | Size of the patch. In case if 100 is specified then empirical local patch will be used | 100 |
| DA\_dir() | HydroDA Directory | $(CaMa-Flood-DA)/$(HydroDA) |
| patch\_dir() | Directory for empirical local patches | $(CaMa-Flood-DA)/ )/$(Empirical-Local-Patch)/local\_patch |
| patch\_name() | Name of the empirical local patch directory | 'amz\_06min\_ERA5\_60' |
| patch\_id() | ID of the empirical local patch (Same as the threshold, but a string) | “0.60” |
| thersold() | threshold for empirical local patch | 0.60 |
| initial\_infl() | initial inflation parameter | 1.08 |
| rho() | inflation parameter | 1.08 |
| sigma\_b() | background variance of inflation for adaptive inflation | 0.04 |
| ens\_mem(mode=mode()) | number of ensemble members | 20 |
| timestep() | outer time step in seconds | 86400 |
| starttime() | start time for data assimilation | (2014,1,1) |
| endtime() | end time for data assimilation | (2015,1,1) |
| spinup\_mode() | decides which experiments should the spinup simulation be done for | 0 (do spinup simulation for both (corrupted/open and true) simulation |
| spinup\_end\_year() | ending year for spinup simulation | 2013 |
| spinup\_end\_month() | ending month for spinup simulation | 12 |
| spinup\_end\_date() | ending date for the spinup simulation | 31 |
| runoff\_dir() | Directory for runoff forcings | $(CaMa-Flood-DA)/$(Ensemble-Simulation)/CaMa\_in/ERA5 |
| runname(num=mode()) | Name of runoff forcing dataset | “ERA5” |
| input(num=mode()) | Name of runoff forcing dataset | “ERA5” |
| run\_flag() | decides which assimilation experiments to run | 0 (run all simulations) |
| CaMa\_dir() | CaMa-Flood Model directory | $(CaMa-Flood-DA)/$(CaMa-Flood) |
| obs\_name() | Name of the observation dataset to be assimilated | “HydroWeb” |
| HydroWeb\_dir() | Location of satellite observation directory | $(CaMa-Flood-DA)/HydroWeb |
| obs\_dir() | Location of processed satellite observations | $(CaMa-Flood-DA)/ $(HydroDA)/obs/HydroWeb |
| obs\_list() | Location of text file describing list of the satellite observations | DA\_dir() + "/dat/HydroWeb\_alloc\_" + mapname() + "\_org.txt" |
| stat\_name(cal=calibrate()) | Name of sfcelv statistics file | "sfcelv\_20\_ERA5\_amz\_06min\_2000-2020" |
| para\_nums() | Number of parallel model runs | 20 |

1. Modify **$(CaMa-Flood-DA)/$(HydroDA)/gosh/run\_mool.sh** specifying HydroDA=$(CaMa-Flood-DA)/ $(HydroDA) and HydroDAout=$(CaMa-Flood-DA)/ $(HydroDA). Also specify the name of the experiment by setting EXP=’test\_wse’ (can be any name).
2. Execute run\_mool.sh by running ‘./run\_mool.sh’. It is preferable to use qsub here if you are using the rainbow server.
3. Outputs of the data assimilation experiments are generated in **$(CaMa-Flood-DA)/$(HydroDA)/out/$(EXP)/assim\_out/**.
4. A timeseries of the discharge output can be plotted by running **p01\_plot\_timeseries.py** in **$(CaMa-Flood-DA)/$(HydroDA)/img.**