APS Weather Model in Python

ASF Engineering

Abstract

This document is a guide to users of the APS Weather Model Correction software. It includes installation and operation instructions as well as examples of code usage.

Usage

The python version of the APS weather model code runs much like the original matlab version. Individual steps in the process can be selected via the start step and end step command line arguments. One additional step was added to the original code – step 4 subtracts the calculated phase delay from the interferograms, assuming the standard ASF naming convention is used for these inputs (date1_date2_unw_phase.tif). Currently, no model type is required on the command line since only the merra2 model is supported. However, internally, the code is setup to be able to plug in new weather model modules without great difficulty.

```
usage: aps_weather_model [-h] [-g GEO_REF_FILE] sstep estep

Calculate tropospheric (slant) delay for a stack of interferograms:
     0 - Set up list of weather data files to download
     1 - Download weather data
     2 - Calculate wet and hydrostatic zenith delays
     3 - Calculate the SAR delays
    4 - Subtract calculated delay from interferograms

positional arguments:
  sstep                 Start step to begin processing (0-4)
  estep                 End step to stop processing (0-4)

optional arguments:
  -h, --help            show this help message and exit
  -g GEO_REF_FILE, --geo_ref_file GEO_REF_FILE
                        Name of file for georeferencing information
```

Installing the aps_weather_model Software

The aps_weather_model python software is available via github. Once you have a github account (https://github.com) you can download the software using these commands:

```
$ git clone git@github.com:asfadmin/hyp3-lib.git
$ git clone git@github.com:asfadmin/hyp3-TRAIN.git
```

Having cloned the software, you could add the following lines to your .bashrc file in order to add the package to your PATH and PYTHONPATH environment variables:

```
export PYTHONPATH=~/hyp3-lib/src
export PATH=~/hyp3-lib/src:$PATH
export PATH=~/hyp3-TRAIN/src:$PATH
```

Access to MERRA2 Weather Model Data

In order to download the MERRA2 weather model data, you must have a NASA Earthdata account. If you do not have an Earthdata Login, you can create one at https://urs.earthdata.nasa.gov/users/new.

Once you have an Earthdata login, you must create a .netrc file in your home directory. The file has this form:

```
machine urs.earthdata.nasa.gov
    login    username
    password password
```

Where, *username* and *password* are your Earthdata login information.

Finally, you must approve the “NASA GESDISC DATA ARCHIVE” application to use your Earthdata login to access the data from the Goddard Earth Sciences Data and Information Services Center (GES DISC):

1) Login to https://urs.earthdata.nasa.gov/home
2) Go to "Applications" → “Authorized Apps”
3) Go to "Approve More Applications" at the bottom of the page
4) Approve an application called NASA GESDISC DATA ARCHIVE

Once you have gone through these steps, the MERRA2 weather model data will be automatically downloaded when using the aps_weather_model.py codes.

Creating a parms_aps.txt file

In order to run the aps_weather model, you must create a parms_aps.txt file. You must also have a reference geotiff or a latitude, longitude matrix. If a geo_ref_file is given on the command line, then the following information is gathered from it (1) region_res, (2) region_lat_range, (3) region_lon_range, (4) latitude, longitude matrix. Otherwise, these parameters must appear in the parms_aps.txt file. Thus, an example input parameter file might contain the following if a geo_ref_file is used with the “asf” options:

```
UTC_sat: 46499.182780
merra2_datapath: /hugeslice/rtc/TOM/TRAIN/TRAIN_TEST/merra
DEM_origin: asf
DEM_file: new_dem.tif
lambda: 0.05546576
incidence_angle: 0.67195
date_origin: asf
```

While, an example parameter file might contain the following if no geo_ref_file is used and no “asf” options are used. In this case, input_dem.tif, ifgday.txt, and llmat.txt must already exist:

```
UTC_sat: 46499.182780
merra2_datapath: /hugeslice/rtc/TOM/TRAIN/TRAIN_TEST/merra
DEM_origin: file
DEM_file: input_dem.tif
lambda: 0.05546576
incidence_angle: 0.67195
date_origin: file
ifgday_file: ifgday.txt
region_lat_range: 31.4660970 33.5199537
region_lon_range: -102.1647111 -105.2093932
region_res: 0.000802287764870
ll_file: llmat.txt
```

Keywords in params_aps.txt file

NAME	REQ ?	TYPE	NOTES
UTC_sat	Yes	Float	UTC time of satellite passes given as seconds of the day
{model}_datapath	Yes	String	Full data path to the weather files. If path doesn't exist, it will be created. Currently only accepts merra2 for model
DEM_origin	Yes	String	One of "asf", "opentopo", or "file". "asf" gets the DEM file from the ASF DEM heap. "opentopo" gets the DEM from the opentopo service.
DEM_file	Yes	String	If DEM_origin is "file", this file name will be used and is assumed to already exist. If the DEM_origin is "opentopo" or "asf", this DEM name will be used to create a new DEM. Note, only geotiff DEM files are currently supported.
lambda	Yes	Float	SAR wavelength
incidence_angle	Yes	Float/ String	Incidence angle of SAR in radians. Either a single floating point value or a file name. The file should contain incidence angles corrected for the curvature of the earth. It must be a geotiff the same size and projection as the interferograms.
region_lat_range	Opt	Float	If no geo_ref_file is given as input on the command line, this range will be used.
region_lon_range	Opt	Float	If no geo_ref_file is given as input on the command line, this range will be used.
region_res	Opt	Float	If no geo_ref_file is given as input on the command line, this resolution will be used. Note, the resolution is in degrees.
date_origin	Yes	String	One of "asf" or "file". If "asf", then dates will be gathered from {date1}_{date2}_unw_phase.tif files.
ifgday_file	Opt	String	If date_origin is "file", this file name will be used to gather the dates of each interferogram to correct.
ll_file	Opt	String	If no geo_ref_file is given as input on the command line, this file will be used for the latitude, longitude grid representing the SAR area of coverage.

Creating an llmat.txt file

This step is only required if you do not have a geo_ref_file. The llmat.txt ASCII text file contains the latitude, longitude coordinates of each of the points in the SAR interferograms. Thus, for example, to create a llmat.txt file using python:

```
>>> import numpy as np
>>> lon = np.zeros((2560,3795))
>>> lat = np.zeros((2560,3795))
>>> for i in range(2560):
...     for j in range(3795):
...         lat[i,j] = 33.519953688252485 + i*-0.000802287764870
...         lon[i,j] = -105.209393171633067 + j*0.000802287764870
...
>>> ll = np.zeros((2560*3795,2))
>>> ll[:,0] = lat.flatten()
>>> ll[:,1] = lon.flatten()
>>> np.savetxt("llmat.txt",ll))
```

The resulting file contains these lines:

```
-105.20939317   33.51995369
-105.20859088   33.51995369
-105.2077886    33.51995369
...,
-102.16711797   31.4668993
-102.16631568   31.4668993
-102.16551339   31.4668993
```

Creating an ifgday.txt file

The ifgday.txt file contain the dates of the interferograms to be corrected. It is in the form [n_ifgs 2]. For example, the following would be a valid ifgday file describing 6 interferogram pairs:

```
20160930 20161024
20160930 20161117
20161024 20161117
20161024 20161211
20161117 20161211
20161117 20170104
```

Using ASF HYP3 Data

The python version of the APS weather model is designed to easily use the data that comes out of ASF's HYP3 website. Several features simplify running the weather model correction procedure:

- The interferogram date file is generated from the geotiff files found in the directory where the weather model correction is run. Setting the "date_origin" keyword to "asf" activates this feature.
- The latitude, longitude matrix (llmat.txt) file is automatically generated using the input geo_ref_file. Using a geo_ref_file on the command line activates this feature.
- The region_res, region_lat_range, and region_lon_range are all automatically set using information from the geo_ref_file. Using a geo_ref_file on the command line activates this feature.
- A DEM can be automatically generated during the run using the ASF DEM heap. This is activated by the "DEM_origin" keyword being set to "asf".

As one can see, the use of the aps_weather_model.py is greatly simplified when using ASF file formats (geotiff) and ASF file names (date1_date2_unw_phase.tif). The UTC time can be garnered from the *.txt metadata files that come with the HYP3 outputs. As for the look angle (really the incidence angle), for Sentinel, it varies from 31 to 46 degrees. The average would be 38.5 degrees, or 0.67195 radians.

Finally, any of the *_*_unw_phase.tif files can be selected for the geo_ref_file – they all should have the exact same size and geographic information provided two final things are known - the inputs to the aps_weather_model.py are (1) assumed to be in geographic coordinates and (2) to be of exactly the same size.

The automatic time series generation using HyP3 InSAR products (not part of HYP3 currently) will clip all files to matching sizes and UTM projections. If, however, INSAR products are ordered individually through HyP3, they will need to be clipped the same size. This can be accomplished using the utility "cutGeotiffs.py" that is included in the hyp3-TRAIN software package:

```
$ cutGeotiffs.py *_unw_phase.tif
$ for file in *_clip.tif; do mv $file ${file%_clip.tif}.tif; done
```

Once files are the same size, the outputs from Gamma INSAR runs will have to be warped from UTM to GCS coordinates. This can be easily accomplished using gdal:

```
$ gdalwarp -t_srs EPSG:4326 in_utm_file out_gcs_file
```