Initiation to SAR interferometry
Basics of ***AMSTerEngine***

Dominique Derauw

September 2021

1 Objective of the document

This document describes the basic operation of ***AMSTerEngine***, outlining the steps of a simple process aimed at creating a digital elevation model or a differential phase measurement using a pair of SAR images. In this case, the steps presented are based on the processing of the SAOCOM test pair available on the CONAE website. (`https://catalogos.conae.gov.ar/catalogo/catalogosatsaocomadel.html`)

2 Command line

The ***AMSTerEngine*** is a set of commands that are launched from a terminal. In general, commands accept one or more parameters. Parameters must be separated by spaces. Parameters can be composed of options (*flags*), in which case the parameter begins with a hyphen followed by the chosen option(s).

When a command is launched without parameters or with the `-h` option, a brief *help* message appears. This *help* message describes the parameters and options that can be used. Parameters and options enclosed in square brackets are optional.

2.1 Environment variables

Some ***AMSTerEngine*** routines use specific directories were they expect to find some dedicated information. While not mandatory, predefining these directories through environment variable setting is of good practice. These variables are the following:

- `EXTERNAL_DEMS_DIR`
- `EXTERNAL_MASKS_DIR`
- `EARTH_GRAVITATIONAL_MODELS_DIR`
- `ENVISAT_PRECISES_ORBITS_DIR`
- `S1_ORBITS_DIR`

2.2 Directory and file path

It must be noted that ***AMSTerEngine*** does not accept folder or file locations that have spaces in their names or anywhere in their path.

3 Data reading

We assume that the data is downloaded and located in the folder:
`${HOME}/SAR/DATA/Images/SAOCOM/TestDataSet`
where the folder `${HOME}` is the user's home directory.

The first step is to read the data and convert it to the `.csl` format used by ***AMSTerEngine***. The data will be saved in the folder `${HOME}/SAR/Processing/SAOCOM/ElLeoncito` to be created beforehand; El Leoncito is the name of the region observed.

Of course, the user can change the names as desired. The reading itself is done using the `SAOCOMDataReader` command.

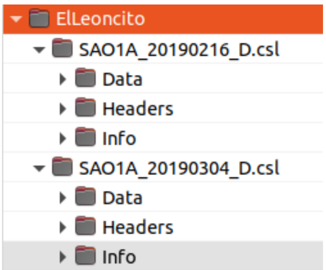
This command accepts two parameters: an input folder and a destination folder. All data found in the input folder will be read and saved in the destination folder.

The location of the input folder is mandatory. The destination folder is optional. If no destination folder is specified, the read data will be saved in the input folder.

Therefore, in the case taken as an example in this document, the command line to use will be:

```
SAOCOMDataReader ${HOME}/SAR/DATA/Images/SAOCOM/TestDataSet
${HOME}/SAR/Processing/SAOCOM/ElLeoncito
```

After running this command, both data sets will be read and will appear in the destination folder in `.csl` format.



3.1 Others sensors

When using images from other satellites, the process is similar, using the commands `CSKDataReader`, `TSXDataReader`, `RSATDataReader`, `ICEYEDataReader`, `NISARDataReader`, `EnviSATBulkDataReader`, and `S1DataReader`.

In the latter case, the `S1DataReader` command accepts an additional optional parameter in the form of a kml file to limit the readout to Sentinel bursts that intersect the area of interest.

4 External DEM management

After reading the data, an external DEM can be associated with the image considered as the master of the interferometric process. ***AMSTerEngine*** accepts external DEMs in ENVI and SRTM DEM formats, provided that the necessary SRTM DEM elements are assembled with ***AMSTerEngine***.

4.1 SRTM DEM

In order to get an SRTM DEM, one can use the `getSRTMDEM` routine. This routine will download SRTM tiles encompassing given kml from NASA servers, aggregate them and save DEM within

DEM directory defined by `EXTERNAL_DEMS_DIR` environmental variable

By default, saved DEM will have the same name than given kml. An alternative name or full path can be given using second parameter.

SRTM tiles are fetched from:

https://e4ftl01.cr.usgs.gov/MEASURES/NASADEM_HGT.001/2000.02.11/

Add your credentials to your .netrc file correspondingly.

If `EXTERNAL_MASKS_DIR` environmental variable is defined, water body mask will also be saved within the corresponding mask directory.

SRTM heights are referenced to Earth Gravity Model EGM96 while ***AMSTerEngine*** make use of ellipsoidal heights. Genereated DEM will thus be "Corrected for" EGM96 geoidal ondulations if available within the `${EARTH_GRAVITATIONAL_MODELS_DIR}/EGM96` directory. EGM96 file can be donloaded at:

`http://download.osgeo.org/proj/vdatum/egm96_15/outdated/WW15MGH.DAC`

Or

`https://github.com/jleppert/egm96/blob/master/WW15MGH.DAC`

(link checked on April 30, 2025)
Or ask it to me politely and I will send you the EGM96 archive I have (dderauw@sareos.be).

The kml file generated at data reading within the ./Info sub directory of .csl image can be used directly to download the corresponding SRTM DEM. In this case the command to be used will be:

```
getSRTMDEM ${HOME}/SAR/Processing/SAOCOM/ElLeoncito/SAO1A_20190216_D.csl/Info/
SAOCOM_1A_20190216_D.kml ElLeoncito
```

If the environment variable `EXTERNAL_DEMS_DIR` is correctly configured, the SRTM DEM encompassing the whole image will be saved at path `${EXTERNAL_DEMS_DIR}/ElLeoncito` with its corresponding descriptive .txt file.

4.2 Association of an external DEM to a `.csl` image

Generally, regardless of the process, we need to use an external DEM to extract the differential phase or to geoproject the generated products. Since the interferometric products to be calculated will all be in the acquisition geometry of the master image, the external DEM must be associated with and projected onto the range-azimuth coordinate system of that image.

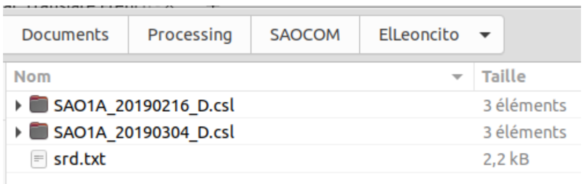
This association is made using the texttts1antRangeDEM command.

This command requires a text file with the characteristics of the process to be performed as a parameter. This file can be created by the command itself using the `-create` option. Therefore, the `slantRangeDEM` command must be used twice: once to create the file in a specific location, and a second time using the file created and modified as required.

4.2.1 Creation and use of the parameter file

In the following, we consider the most recent SAOCOM image as the ***Master*** and we will create the parameter .txt file within the directory where this image is located. Name of the text file is at user's choice. Used command can be for example:

`slantRangeDEM ${HOME}/SAR/Processing/SAOCOM/ElLeoncito/srd.txt -create`




```
/* Slant range DEM generation */
/* ************************** */
/home/dd/Documents/Processing/SAOCOM/ElLeoncito/SAO1A_20190304_D.csl /* Reference slant range image path in CSL format */

/* External DEM characteristics */
/* **************************** */
ElLeoncito                          /* Georeferenced DEM file path */
7201                                /* X (longitude) dimension [pixels] */
7201                                /* Y (latitude) dimension [pixels] */
-70                                 /* Lower left corner longitude [dd] */
-33                                 /* Lower left corner latitude [dd] */
0.0002777777777777778               /* Longitude sampling [dd] */
0.0002777777777777778               /* Latitude sampling [dd] */
NaN                                 /* Excluding value */

/* Resampling parameters */
/* ********************* */
TRI                                 /* Resampling method : TRI = Triangulation; AV = weighted average; NN = nearest neighbour */
LORENTZ                             /* Weighting method : ID = inverse distance; LORENTZ = lorentzian */
1.0                                 /* ID smoothing factor */
1.0                                 /* ID weighting exponent */
0.7                                 /* FWHM : Lorentzian Full Width at Half Maximum */

/* Reduction factors */
/* ***************** */
3                                   /* Range reduction factor */
6                                   /* Azimuth reduction factor */

/* Mask characteristics */
/* ******************** */
File path                           /* Georeferenced mask file path */
1000                                /* Mask X (longitude) dimension [pixels] */
1000                                /* Mask Y (latitude) dimension [pixels] */
5.5655                              /* Mask lower left corner longitude [dd] */
50.5980                             /* Mask lower left corner latitude [dd] */
0.00027777777                       /* Mask longitude sampling [dd] */
0.00027777777                       /* Mask latitude sampling [dd] */
```

This file must be changed manually to correctly project the external DEM onto the coordinate system of the master image (these changes are made automatically when using the `lazInSAR` command).

The parameter in line 3 must be replaced with the location of the master image we are referring to; in this case:

`${HOME}/SAR/Processing/SAOCOM/ElLeoncito/SAO1A_20190304_D.csl`

The parameter in line 7 must be replaced with the name of the DEM to be used; in this case: `ElLeoncito`, provided the `EXTERNAL_DEMS_DIR` environment variable is correctly defined.

If this is not the case, the full path to the DEM must be provided. In this case, it would be: `${HOME}/SAR/DATA/DEMS/ElLeoncito`

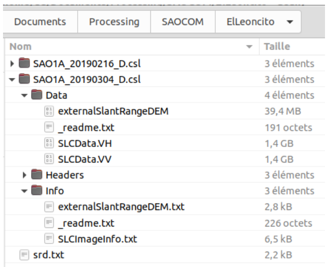
If you already know what reduction factors will be used in the interferometric process, you can also change the corresponding parameters in lines 26 and 27 of the file.

The other parameters can remain unchanged.

After saving and closing the interface file, the external DEM is projected onto the coordinate system of the ***Master*** image by repeating the same command without the option. `-create`:

`slantRangeDEM ${HOME}/SAR/Processing/SAOCOM/ElLeoncito/srd.txt`

By doing so, the modified interface file `srd.txt` will be used by the `slantRangeDEM` command and, at the end of the process, the reference DEM projected in *range* - *azimuth* coordinates of the ***Master*** image will be created and saved in the `./DATA` folder of the ***Master*** `.csl` image.



5 Interferometric processing

5.1 Initialization

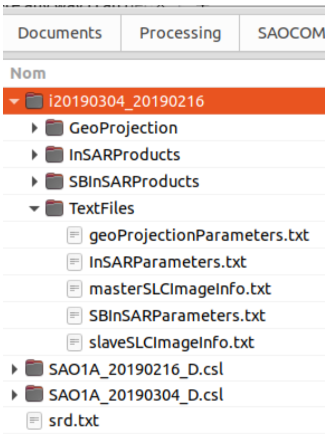
Now we are ready to perform an interferometric process. First, we must create the working folder with all the necessary structure and interface files. This is done using the `initInSAR` command.

This command requires at least two parameters: the ***Master*** image and the ***Slave*** image. So, in our case, the command to use will be:

```
initInSAR ${HOME}/SAR/Processing/SAOCOM/ElLeoncito/SAO1A_20190304_D.csl
${HOME}/SAR/Processing/SAOCOM/ElLeoncito/SAO1A_20190216_D.csl
```

Using this command will create a process folder called `i20190304_20190216` inside the folder where the ***Master*** image is located. If you prefer another name and/or another location for your working folder, you must specify that location as an additional parameter. For example:

```
initInSAR ${HOME}/SAR/Processing/SAOCOM/ElLeoncito/SAO1A_20190304_D.csl
${HOME}/SAR/Processing/SAOCOM/ElLeoncito/SAO1A_20190216_D.csl
${HOME}/SAR/Processing/SAOCOM/ElLeoncito/MyFirstAMSTerEngineInSARProcessing
```



As shown in the figure, initialization creates a subfolder structure that will containe the interferometric products and the geoprojected products. The `TextFiles` folder contains the information files relating to the ***Master*** image and the ***Slave*** image, as well as the `InSARParameters.txt` and `geoProjectionParameters.txt` interface files for the interferometric and geoprojection processes.

These interface files contain useful information (interferometric baseline size, ambiguity height at the center of the image, etc.) as process parameters. This information can be used to select image pairs offering the most suitable interferometric baseline for your process.

```
/* Slant range DEM generation */
/* ************************** */
/home/dd/Documents/Processing/SAOCOM/ElLeoncito/SAO1A_20190304_D.csl /* Reference slant range image path in CSL format */

/* External DEM characteristics */
/* **************************** */
ElLeoncito                          /* Georeferenced DEM file path */
7201                                /* X (longitude) dimension [pixels] */
7201                                /* Y (latitude) dimension [pixels] */
-70                                 /* Lower left corner longitude [dd] */
-33                                 /* Lower left corner latitude [dd] */
0.00027777777777777778              /* Longitude sampling [dd] */
0.00027777777777777778              /* Latitude sampling [dd] */
NaN                                 /* Excluding value */

/* Resampling parameters */
/* ********************* */
TRI                                 /* Resampling method : TRI = Triangulation; AV = weighted average; NN = nearest neighbour */
LORENTZ                             /* Weighting method : ID = inverse distance; LORENTZ = lorentzian */
1.0                                 /* ID smoothing factor */
1.0                                 /* ID weighting exponent */
0.7                                 /* FWHM : Lorentzian Full Width at Half Maximum */

/* Reduction factors */
/* ***************** */
3                                   /* Range reduction factor */
6                                   /* Azimuth reduction factor */

/* Mask characteristics */
/* ******************** */
File path                           /* Georeferenced mask file path */
1000                                /* Mask X (longitude) dimension [pixels] */
1000                                /* Mask Y (latitude) dimension [pixels] */
5.5655                              /* Mask lower left corner longitude [dd] */
50.5980                             /* Mask lower left corner latitude [dd] */
0.00027777777                       /* Mask longitude sampling [dd] */
0.00027777777                       /* Mask latitude sampling [dd] */
```

The interface files are prepared using a standard process, so they can be used without changing anything.

Below we will look at some of the simplest parameters that can be changed to customize the processing.

5.2 General use of *AMSTerEngine* commands

The different commands that will be used to perform the interferometric process require the interface file `InSARParameters.txt`, so that file must be found by each command. This can be done by providing the full path to the file as a parameter to the commands in question or, more easily, by changing your working directory in the terminal using the command `cd`.

```
dd@SAREOS:~$
dd@SAREOS:~$ cd '/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304_20190216'
dd@SAREOS:~/Documents/Processing/SAOCOM/ElLeoncito/i20190304_20190216$
```

By doing so, all commands will be launched from the working folder ${HOME}/SAR/Processing/SAOCOM/ElLeoncito/i20190304_20190216 and the interface file will be found automatically without the need to specify the path as a parameter. In this document, we assume that the user has changed their working folder to that of the interferometric process.

5.3 Corregistration

With stripmap images, corregistration is performed in four steps:

- `amplitudeImageReduction`
- `coarseCoregistration`

- `fineCoregistration`
- `interpolation`

With Sentinel-1 TOPSAR images, corregistration is performed in a single step using the `S1Coregistration` command.

5.3.1 `amplitudeImageReduction`

The first step consists of generating reduced images in modules, averaging pixels over windows of specific dimensions. If we look at the interface file `InSARParameters.txt`, we see that the predefined reduction parameters are 3 x 6 for SAOCOM images. The size of the reduction window is chosen to generate approximately square pixels in flat areas by averaging at least 10 pixels. The aim is to obtain reduced images with a good signal-to-noise ratio, allowing the images to be correlated at anchor points in the coarse co-registration step.

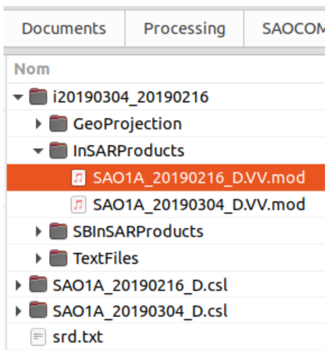
```
/* ******************** */
/* * InSAR processing * */
/* ******************** */

/* -1- Box averaging  */
/* ***************** */
/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304_20190216/InSARProducts/SAO1A_20190304_D.VV.mod /* Reduced master amplitude image file path */
2347                                    /* Reduced master amplitude image range dimension [pix] */
3944                                    /* Reduced master amplitude image azimuth dimension [pix] */
/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304_20190216/InSARProducts/SAO1A_20190216_D.VV.mod /* Reduced slave amplitude image file path */
2347                                    /* Reduced slave amplitude image range dimension [pix] */
3944                                    /* Reduced slave amplitude image azimuth dimension [pix] */
3                                       /* Range reduction factor [pix] */
6                                       /* Azimuth reduction factor [pix] */
```

The location and name of the resulting files are predefined. These parameters can be changed as desired. The same applies to the reduction parameters. Dimensions of the images will be updated in the text file at the end of the process.

The command to use is simply:

`amplitudeImageReduction`

The reduced images will appear as the first interferometric products within the folder. `./InSARProducts`.



5.3.2 `coarseCoregistration`

Now that we have the images reduced in module, we can proceed to coarse co-registration, looking for anchor points that correspond in both images. This is done by cutting windows of dimensions equal to a power of 2 around the candidate anchor point in the master and slave and correlating them to find the displacement that allows these windows to be superimposed with greater precision. The windows are shifted by a certain number of pixels in range and azimuth within the reduced images. This shift determines the number of candidate anchor points. For each window pair, the correlation results in a peak whose center gives the translation to be applied to the ***Slave*** window so that it overlaps with the ***Master*** window. The choice of an anchor point as good is made by considering three parameters that can be adapted:

- The maximum correlation, which is a number between 0 and 1. By default, the threshold is 0.5, rejecting points for which the correlation does not reach that level.
- The width of the correlation peak in both dimensions. By default, the width at half height of the pixel must be less than 5 pixels in both dimensions. Points for which the correlation peak is wider are rejected.

  The command to use is `coarseCoregistration` and does not require parameters, as these will be read from the parameter file.

  At the end of processing, an affine transformation is calculated that allows the coordinates of the points in the ***Slave*** image to be related to those in the ***Master*** image.

```
/* -2- Coarse coregistration */
/* ************************ */
64                                  /* Coarse coregistration range window size [pix] */
64                                  /* Coarse coregistration azimuth window size [pix] */
32                                  /* Coarse registration range distance between anchor points [pix] */
32                                  /* Coarse registration azimuth distance between anchor points [pix] */
0.5                                 /* Coarse coregistration correlation threshold */
5                                   /* Corelation peak range width threshold */
5                                   /* Corelation peak azimut width threshold */

/* -3- Fine coregistration */
/* ********************** */
5                                   /* Fine coregistration range window size [pix] */
11                                  /* Fine coregistration azimuth window size [pix] */
3                                   /* Estimated range error [pix] */
3                                   /* Estimated azimuth error [pix] */
20                                  /* Fine registration range distance between anchor points [pix] */
20                                  /* Fine registration azimuth distance between anchor points [pix] */
0.5                                 /* Fine coregistration correlation threshold */

/* -4- Interpolation */
/* **************** */
/* Master to slave affine coordinate transform: */
/* x2 = Ax x1 + Bx y1 + Cx */
/* y2 = Ay x1 + By y1 + Cy */
0.9993263149401678                  /* Ax */
4.374940454204052e-05               /* Bx */
-109.2456737540247                  /* Cx */
-0.0001754927595146109              /* Ay */
0.9999988559202363                  /* By */
-17.6304952431259                   /* Cy */
/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304_20190216/InSARProducts/SAO1A_20190216_D.interpolated.csl /* Interpolated slave image file path */

/* Global master to master InSAR processing */
NONE                                /* Global master to master InSAR directory path */
```

5.3.3 `fineCoregistration`

The fine co-registration stage is based on the previously obtained affine transformation and aims at improving it by optimizing coherence at the local level. Therefore, the process is similar to the previous one, but here the windows are smaller and are cut out from complex high-resolution images (`SLC`). In addition, they are shifted by default by 20 pixels in each dimension, so the number of candidate anchor points is much larger than for coarse co-registration. Each slave window will move relative to the master window by a certain number of pixels in both dimensions. By default, the shift values for the slave window are +/-3 pixels in both dimensions. For each displacement, a mini interferogram and a corresponding coherence are calculated, so that a coherence map is established around the initial search position. With a displacement of +/-3 pixels, the local coherence map that is established is 7 x 7 pixels. In this map, a peak of maximum coherence that approximates a Gaussian is searched for. The location of the Gaussian maximum corresponds to the co-registration value of the considered anchor point. In the case of fine co-registration, a candidate anchor point is considered as coregistered if local coherence maximum is above a given threshold. By default, the threshold is 0.5.

The command to be launched is `fineCoregistration` and does not require any parameters, as these will be read from the interface file.

At the end of the process, the affine transformation will be updated.

The co-registration steps can be repeated iteratively to improve the results.

For the two co-registration steps, the standard deviation between the calculated affine transformation and all selected anchor points is calculated.

For fine co-registration, an estimate of the interferometric baseline is also calculated based solely on the fine transformation found and the orbit of the master image. This estimate provides an idea of the co-registration accuracy by comparing it with the interferometric baseline calculated using the two orbits.

5.3.4 `interpolation`

After coarse and fine co-registration, we have an affine transformation that allows us to transform the ***Slave*** image to superimpose it on the ***Master*** image. This step is done by complex interpolation.

The command to use is `interpolation`.

The transformation to be used will be read from the interface file. The interpolated ***Slave*** image will be saved within the interferometric products in the `.csl` format.

5.3.5 `S1Coregistration`

In the case of Sentinel1 TOPSAR images, data co-registration and even interpolation are performed using a single command: `S1Coregistration` Co-registration and interpolation are performed burst by burst based on the orbits included in the images. It is therefore best to keep your orbit database up to date with precise and restituted orbits so that they are correctly included in the images when read. This command accepts, among others, the parameter `-A` to force the co-registration and interpolation of all polarizations available in the images. This allows you to choose different polarization configurations when performing an interferometric process.

✎ **Sentinel1 orbit management** ***AMSTerEngine*** manages Sentinel1's precise orbits (*`AUX_POEORB`* Precise orbit products) and restituted orbits (*`AUX_RESORB`* Restituted orbit products). Precise orbits are available 21 days after acquisition, while restituted orbits are available 3 hours later.

When Sentinel1 data is downloaded, the images contain predicted orbit data. These orbits are not accurate enough to perform correct interferometric processing using the phase. If the objective is to use coherence, the predicted orbits are sufficient.

Therefore, when the objective is to perform interferometric processing, the predicted orbits must be replaced by the restituted or precise orbits. This is done automatically when reading the data using `S1DataReader`, provided that the local database with the orbits is up to date.

To this end, the tool called `updateS1Orbits` included in ***AMSTerEngine*** is used. This tool automatically searches the ESA website for the required orbits. If you prefer to use the Alaska SAR Facility website, you must use the `-ASF` option and have a valid account. To gain automatic access and grant the `updateS1Orbits` tool access to the site, you must first follow the instructions found here:

`https://wiki.earthdata.nasa.gov/display/EL/How+To+Access+Data+With+cURL+And+Wget`

5.4 `InSARProductsGenberation`: Interferometric products computation

Now that we have a co-registered image pair, a pixel at a given azimuth-range coordinate shows the same ground surface element in both images. We can then compute the interferometric products, i.e. the images in modulus, the interferogram itself, and the coherence image.

The command to use is: `InSARProductsGeneration`

The calculation parameters found in the interface file that can be modified by the user are:

- The size of the reduction window to improve the signal-to-noise ratio. This size is defined in the same way as for the reduction of amplitude images.
- The size of the moving calculation window on which coherence and interferometric phase are estimated.
- A parameter called `Square spiral size [pix]`. This parameter defines a square window over which the phase is unwrapped allong a square spiral to estimate in advance the local phase slope in order to subtract it from the calculated interferogram and thus improve the estimation of local coherence.

At the end of the process, the dimensions of the interferometric products will be updated, all of them now having the same dimensions.

```
/* -5- Interferometric products computation */
/* **************************************** */
2347                                            /* Interferometric products range dimension [pix] */
3944                                            /* Interferometric products azimuth dimension [pix] */

/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304 20190216/InSARProducts/interfero.VV-VV        /* Interferogram file path */
/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304 20190216/InSARProducts/interfero.f.VV-VV      /* Filtered interferogram file path */
1                                               /* Range filter Full Width at Half Maximum */
1                                               /* Azimuth filter Full Width at Half Maximum */
1                                               /* Power spectrum filtering factor (for adaptative filtering) */

/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304 20190216/InSARProducts/coherence.VV-VV        /* Coherence file path */
3                                               /* Coherence estimator range window size [pix] */
6                                               /* Coherence estimator azimuth window size [pix] */
5                                               /* Square spiral size [pix] */

/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304 20190216/InSARProducts/firstPhaseComponent.VV-VV      /* First phase component file path */
/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304 20190216/InSARProducts/residualInterferogram.VV-VV        /* Residual interferogram file path */
NaN                                             /* Excluding value */
```

As seen in the above figure, some additional products where also calculated:

- `externalSlantRangeDEM` : a copy of the external DEM associated with the ***Master*** image resampled to the size of the interferometric products..
- `topographicPhaseComponent` : the topographic phase computed from the previous file `externalSlantRangeDEM`.
- `residualInterferogram` : As its name suggests, this is the phase that remains after subtracting the topographic phase from the calculated interferogram. If there are no atmospheric disturbances and if the external DEM allows for the calculation of a good topographic phase, this layer is the differential interferogram showing local movements, if any.

5.4.1 `interferogramFiltering`

The interferogram and the residual phase can be filtered to improve the signal-to-noise ratio. Two filters are available: a home-made one and an adaptive Goldstein filter.

The home-made filter applies Gaussian filtering to the real and imaginary parts of the phase before recalculating the filtered phase. Thus, all parts of the interferogram are filtered in the same way regardless of local coherence. The parameters governing the filtering are the range and azimuth full width at half maximum of the Gaussian used for filtering. Width unit is the interferogram pixel. These width values can be fractional.

The Goldstein filter is adaptive. This means that the filtering depends on the local coherence, filtering more where the coherence is low and less where the coherence is high. The `Power spectrum filtering factor` parameter allows you to control the Goldstein filter. If it is 1, the filter is applied as defined in the litterature. With values greater than one, the filtering is weaker. With values less than one, the filtering becomes stronger, although it does not change much.

By default, when using `interferogramFiltering`, the Goldstein filter is applied. To use the home-made filter, set the `Power spectrum filtering factor` parameter to 0.

In addition, using the `-d` option, the home-made filter can be applied before the adaptive filter, thus providing double filtering.

By default, it is the residual interferogram if present that wil be filtered. In which case, if willing to filter the interferogram itself, use the `-F` option.

5.4.2 `phaseUnwrapping`

Phase unwrapping is performed using the `phaseUnwrapping` command.

Two phase unwrapping methods are available. The first is a home-made *branch cut* method, which searches for and connects the residues found in the interferogram to be unwrapped. The second method is the well-known `snaphu`.

Whichever method is chosen, by default, it is the **unfiltered** interferogram that is unwrapped. To force the filtered phase to be unwrapped, use the `-F` option.

If the residual interferogram is present (filtered if the `-F` option is used), it will this one that will be unwrapped.

At the end of the unwrapping, the unwrapped residual phase will be added to the topographic phase to obtain the complete unwrapped phase and convert it into local height.

In the case of differential interferometry, the unwrapped residual phase should not be added to the topographic phase. To force this behavior, use the `-r` option. In this case, the unwrapped phase will be converted into displacement towards the satellite between the acquisition date of the ***Master*** and that of the ***Slave***.

So, it is the `-r` option that controls whether classical or differential interferometry is performed.

✎ **snaphu** By default, the homemade method is used. To force the use of `snaphu`, you must use the corresponding option, i.e., `--snaphu`.

Consequently, the command to use for classical interferometry will be `phaseUnwrapping -F --snaphu` and `phaseUnwrapping -rF --snaphu` for differential interferometry.

Users familiar with `snaphu` can first use the command `phaseUnwrapping --snaphu -init [-Fr]` to generate a `snaphu.conf` file to be modified to correspond to the desired process. A second use of the command without the `-init` option will use the modified `snaphu.conf` file.

✎ **Use of the home-made *branch cut* algorithm** The homemade unwrapping process is obtained using the command `phaseUnwrapping -F` or `phaseUnwrapping -rF`. In this case, the method is controlled by the parameters in the corresponding section of the `InSARParameters.txt` file.

```
/* -6- Phase unwrapping */
/* ******************* */
/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304 20190216/InSARProducts/biasedCoherence.VV-VV      /* Biased coherence file path */
0                                         /* Biased coherence range dimension [pix] */
0                                         /* Biased coherence azimuth dimension [pix] */
5                                         /* Biased coherence estimator range window size [pix] */
5                                         /* Biased coherence estimator azimuth window size [pix] */
5                                         /* Biased coherence square spiral size [pix] */

/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304 20190216/InSARProducts/residus.VV-VV      /* Residus image file path */
0                                         /* Residus image range dimension [pix] */
0                                         /* Residus image azimuth dimension [pix] */

/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304 20190216/InSARProducts/connexions.VV-VV      /* Connexions image file path */
0                                         /* Connexions image range dimension [pix] */
0                                         /* Connexions image azimuth dimension [pix] */
2                                         /* Range minimum search radius [pix] */
2                                         /* Azimuth minimum search radius [pix] */
1                                         /* Range search radius step [pix] */
1                                         /* Azimuth search radius step [pix] */
1000                                      /* Coherence scaling */
3                                         /* Connexion process mode */
0.25                                      /* Coherence cleaning threshold */
0.1                                       /* False residue coherence threshold */
```

The process is carried out automatically in four steps. The first one consists in searching for residues among the pixels. The second step consists in generating a biased and more contrasted coherence map that allows low-coherence paths to be drawn to connect the residues. This step can be controlled by the size of a moving window used to calculate this biased coherence. The third step consists of connecting the residues. The last step consists of using the connections to unwrapp the phase by integration. In general, the default parameters give satisfactory results in most cases, both for the generation of biased coherence and for the connection of residues.

5.5 `geoProjection`

The last step in the process is to geoproject the data. This is done using the `geoProjection` command. This command uses an additional interface file logically named `geoProjectionParameters.txt`

Through this interface file, you can choose

- The type of projection in **UTM** or **GEC** coordinates, where **GEC** stands for geographic coordinates *longitude - latitude*.
- The Geoid onto which the projection is made.
- The sampling of the final product in seconds when using `GEC` projection or in meters when using `UTM` projection.
- The parameters that govern the resampling process.
- The boundaries of the area to be projected. This frame can be defined by the coordinates of its *lower-left* and *upper-right* corners or by providing a kml file from which the circumscribed rectangle will be calculated.
- The list of products to be geo-projected.

```
/* Geoprojection parameters file */
/* ***************************** */

/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304_20190216/TextFiles/InSARParameters.txt /* InSAR parameters file */
UTM                                     /* Chosen projection (UTM or GEC) */

/* Projection ellipsoid */
/* ******************** */
WGS84                                   /* Ellipsoid ID */
6378137.0                               /* Semi major axis */
6356752.314245179295540                 /* Semi minor axis */
0                                       /* X0 */
0                                       /* Y0 */
0                                       /* Z0 */

/* Geographical projection parameters (GEC) */
/* ********************************** */
2                                       /* Longitude sampling [arc second] */
2                                       /* Latitude sampling [arc second] */

/* UTM projection parameters */
/* ************************* */
30                                      /* Easting sampling [m] */
30                                      /* Northing sampling [m] */

/* Resampling parameters */
/* ********************* */
TRI                                     /* Resampling method : TRI = Triangulation; AV = weighted average; NN = nearest neighbour */
LORENTZ                                 /* Weighting method : ID = inverse distance; LORENTZ = lorentzian */
1.0                                     /* ID smoothing factor */
1.0                                     /* ID weighting exponent */
0.5                                     /* FWHM : Lorentzian Full Width at Half Maximum */
mask                                    /* Masking: Use of slant range mask or zoneMap for measurement geo-projection ([mask] or [zoneMap]) */
1                                       /* Zone index if [zoneMap] is selected for masking */
NaN                                     /* No data value */
NaN                                     /* Excluding height value */
-400.                                   /* Minimal height value */
8000.                                   /* Maximal height value */

/* Geoprojection frame */
/* ******************* */
/* Area of Interest can be given as a kml file or be predefined in geoProjection units. (kml prevails) */
pathToKmlFile                           /* Path to a kml file defining the geoProjection area */
445080                                  /* xMin */
513450                                  /* xMax */
6439110                                 /* yMin */
6525540                                 /* yMax */

2280                                    /* X size of geoprojected products [pix] */
2882                                    /* Y size of geoprojected products [pix] */

/* Geoprojected result files */
/* ************************* */
/* YES/NO flags can be set here to select products to be geoprojected */
YES                                     /* Use external DEM as reference (if NO, the computed InSAR DEM will be used */
YES                                     /* Geoproject measurement (slant range topography or deformation map) */
YES                                     /* Geoproject master amplitude */
YES                                     /* Geoproject slave amplitude */
YES                                     /* Geoproject coherence */
NO                                      /* Geoproject interferogram */
NO                                      /* Geoproject filtered interferogram */
YES                                     /* Geoproject residual interferogram */
YES                                     /* Geoproject unwrapped phase */
/home/dd/Documents/Processing/SAOCOM/ElLeoncito/i20190304_20190216/GeoProjection /* Geoprojected products directory */
```

By default, ***AMSTerEngine*** always works in Cartesian coordinates, i.e., with the origin $(0; 0)$ at the bottom left of the coordinate system. Since all computer display systems, including GIS, consider the origin to be at the top left corner of the screen, the results must be flipped vertically. This is done using the `-r` option. In this way, the geoprojected results will be created in the `ENVI` format with their associated descriptive `.hdr` files so that they can be used with any GIS.