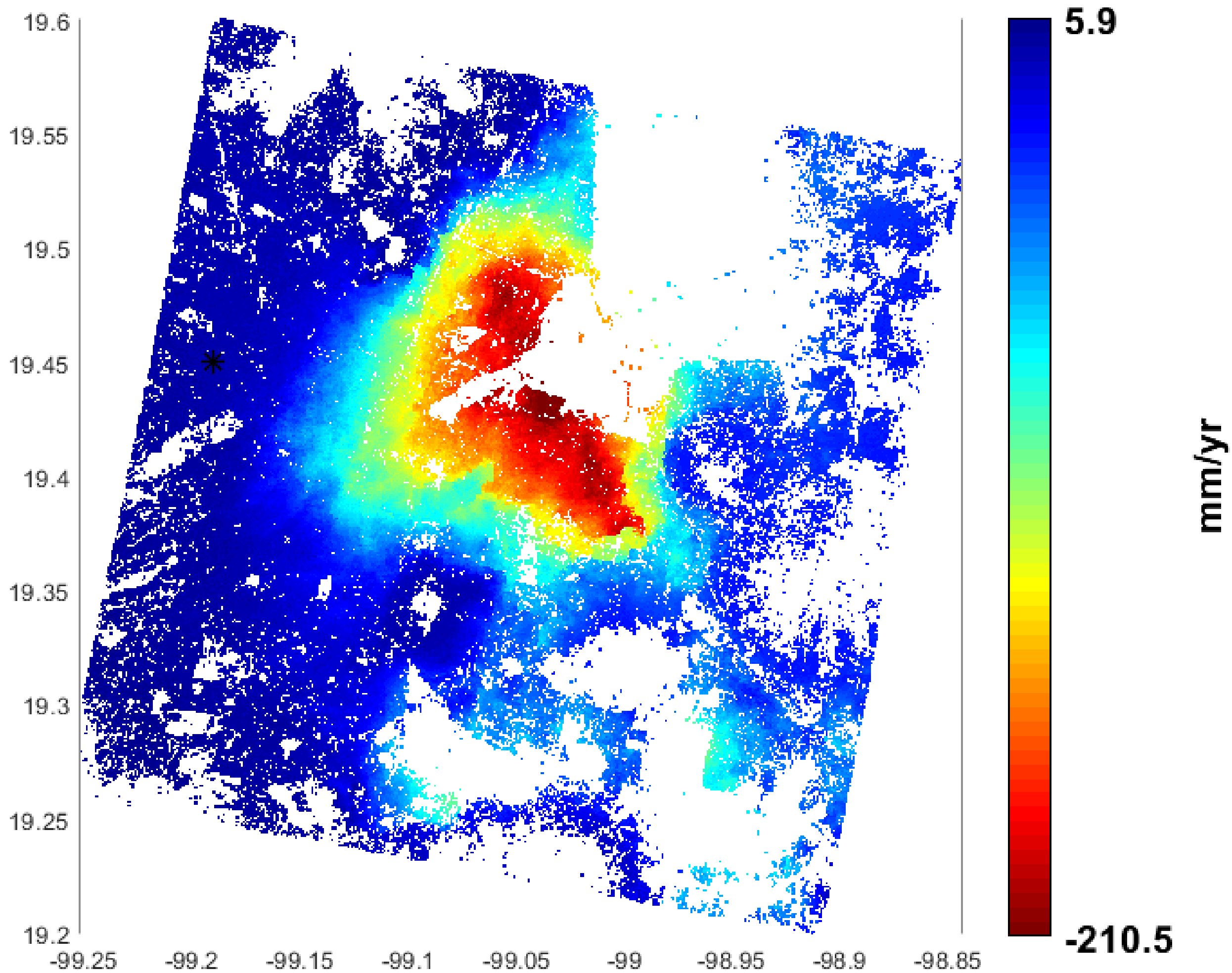
Persistent Scatterer with StaMPS: Mexico city site

Dinh HO TONG MINH
dinh.ho-tong-minh@inrae.fr
Montpellier, July 2021

Goal: Be able to make this final velocity product



This exercise consists of working through an example data set acquired by the Sentinel-1A/B satellite over Mexico city between January 2017 and December 2019. StaMPS can be used for PS or small baseline analysis (or both combined) but in this exercise you will use StaMPS to carry out *PS processing only*.

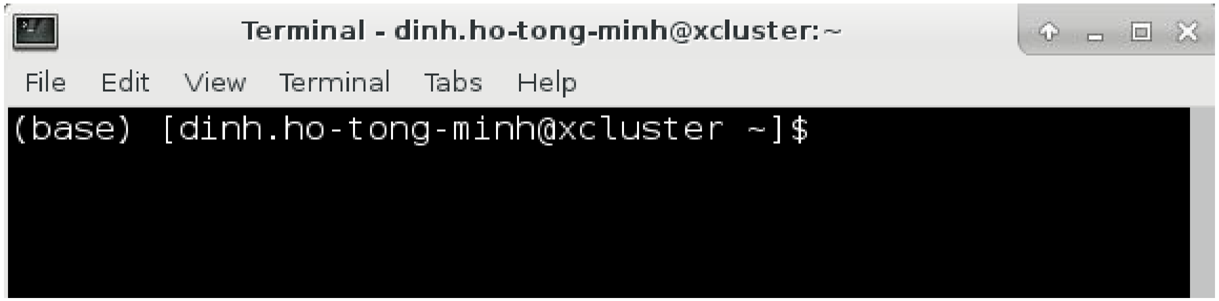
Note that, the data you will be using for this exercise is only a subset of the complete Sentinel-1 acquisitions for this time period. This is so we can finish the analysis within the allocated time and show you the main processing steps.

The StaMPS manual can be found at:
https://github.com/dbekaert/StaMPS/blob/master/Manual/StaMPS_Manual.pdf

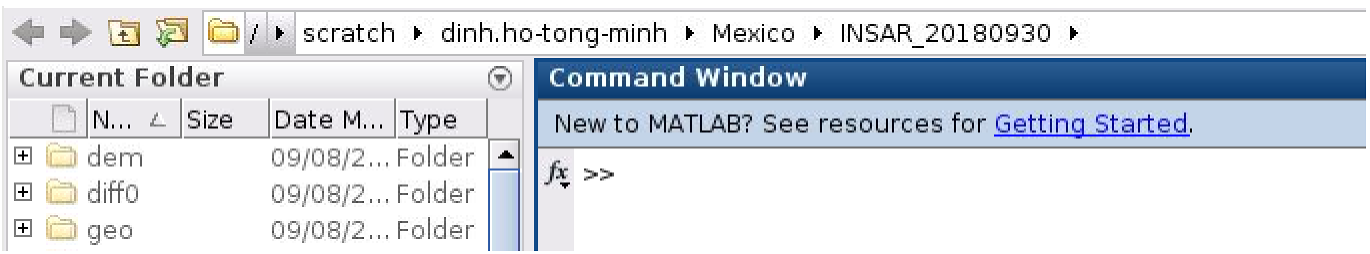
StaMPS also has an active Google Group discussion board where most questions and issues are addressed. If you come across any problems then this would be the first place to check if anyone else has a solution to it.
https://groups.google.com/forum/#!forum/mainsar

In this tutorial the commands in **red** are the ones you will need to run on your computer.
The commands following a:

➢ will need to be run in a linux terminal

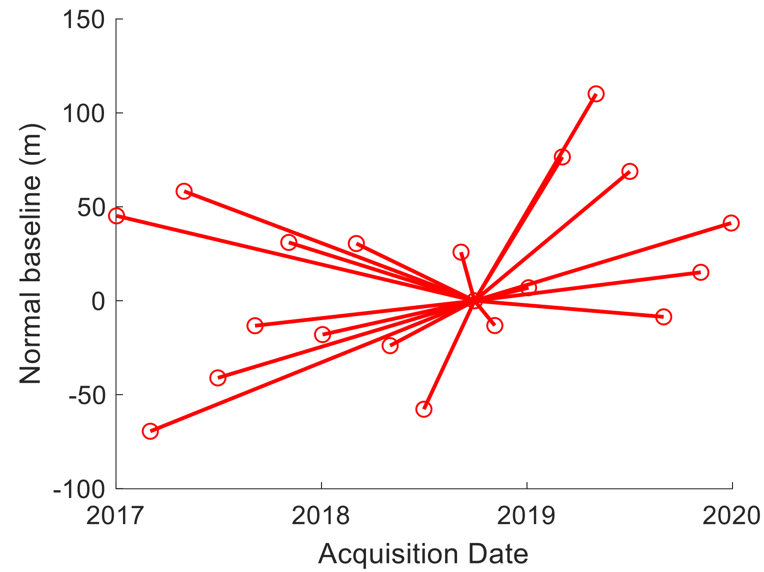


while the ones with **>>** are run in Matlab.



Pre-processing

For this tutorial, we will use 19 single master interferograms that have already been processed with SNAP. The master date for this network is 20180930 (30 September 2018).



First set up your linux environment
(please correct according to your computer):

- export PATH=$PATH::~/occitanie_insar/StaMPS/bin
- export STAMPS=~/occitanie_insar/StaMPS
- export MATLABPATH=~/occitanie_insar/StaMPS/matlab

Let's move to the InSAR processing directory:

- **cd ~/scratch/Mexico/INSAR_20180930**

Candidate PS pixels are selected from the interferograms using:

- **mt_prep_snap 20180930 ~/scratch/Mexico/INSAR_20180930 0.25**

(see beginning of Chapter 6 in the StaMPS manual for details). The pixels selected have amplitude dispersion < 0.25, where amplitude dispersion is the ratio of amplitude standard deviation to mean amplitude. For computing efficiency, you can choose to divide the region in the interferogram into overlapping patches, each of which are processed independently for the initial 5 steps. In this case we have selected to do the processing on a single patch.

Now we can get going with the time series analysis in Matlab.

- **matlab &**

PS Selection

Load data

Step 1 in StaMPS loads in the interferogram data into matlab

>> stamps(1,1)

The two parameters (1,1) given with the stamps command represent the step you want to start with (first parameter) and the step you want to end with (second parameter). The above command will thus only run the 1^{st} step.

(note: check file inputs in *.in and rsc.txt if Matlab can not find the correct files.)

Calculate temporal coherence

This step estimates the spatially correlated phase for each PS candidate, subtracts it, estimates the spatially-uncorrelated DEM error for each pixel from the remaining phase, subtracts this and finally estimates the temporal coherence for each pixel from the residual phase. This process is iterated three times by default. To run this step enter:

>> stamps(2,2)

PS Selection

Initial PS selection

This step makes a selection of PS based on probability, by comparison to results for data with random phase. This is usually done twice. After the first selection, the temporal coherence of each selected pixel is re-estimated more accurately, by dropping the pixel itself from the estimation of the spatially-correlated phase. Then the selection process is repeated.

>> stamps(3,3)

PS weeding

In Step 4, some PS pixels selected in the previous step are dropped based on their proximity to each other and the smoothness of their deformation in time. However, by default the proximity weeding is turned off (‘weed_neighbours’ = ’n’).

>> stamps(4,4)

The main parameter that controls the weeding is the ‘weed_standard_deviation’. This is by default set to 1.

You can list all of the StaMPS parameters by entering:

>> getparm

You can set a parameter (i.e., weed_standard_deviation) by typing:

>> setparm(‘weed_standard_deviation**’, 0.8)**

PS Selection

Phase correction, downsampling and merge

The phase of the remaining selected pixels is corrected for spatially-uncorrelated DEM error and stored as ‘version 2’ (PS candidates were version 1).

>> stamps(5,5)

If you have multiple patches Step 5 also merges them all together into a single dataset.

Downsampling points

By default, PS points are retained at full resolution. However, for many applications, we do not need such a high sampling rate in space. The data can be resampled to a coarser grid at this Step. An additional effect of the resampling is to reduce the noise, and resampling also offers another chance to reject the noisiest pixels.

To change the downsampling, set the parameter ‘merge_resample_size’ (units are m, default is 0, meaning no resampling) and rerun Step 5. Check how the wrapped phase looks now with ps_plot. Try also varying ‘merge_standard_dev’ (threshold standard deviation in radians) and check the effect that it has.

To check the baselines and PS, enter **>> ps_info**

Phase unwrapping

Once you are happy with the PS selection you can unwrap the phase using StaMPS Step 6. During unwrapping, the data are temporarily resampled to a grid. The size of the grid is controlled by parameter ‘*unwrap_grid_size*’, and should be at least as large as ‘*merge_grid_size*’. Go ahead and set it to the same value, then run Step 6.

```
>> stamps(6,6)
```

Once Step 6 is finished, plot the results using:

```
>> ps_plot('u')
```

Try plotting an individual interferogram, e.g. to plot the tenth one:

```
>> ps_plot(‘u',1,0,0,10)
```

The main parameters to play with to improve unwrapping are ‘*unwrap_grid_size*’ – making it larger reduces noise but it may also alias (undersample) the signal, and ‘*unwrap_time_win*’, which is the filter length (in days), used to estimate the noise contribution for each phase arc (phase difference between neighboring PSs).

Estimation of spatially-correlated errors

Step 7 estimates the total DEM error, and the master atmosphere and orbit error. Run Step 7

>> stamps(7,7)

Removing the spatially-correlated error from the wrapped phase can actually help with the unwrapping, and it is a good idea to run Step 6 again after running Step 7 (estimates from Step 7 are automatically subtracted from the wrapped phase before unwrapping).

For example, **>> setparm('scla_deramp', 'y')** and **>> stamps(6,7**)

Velocity estimation

You can plot the mean velocity using

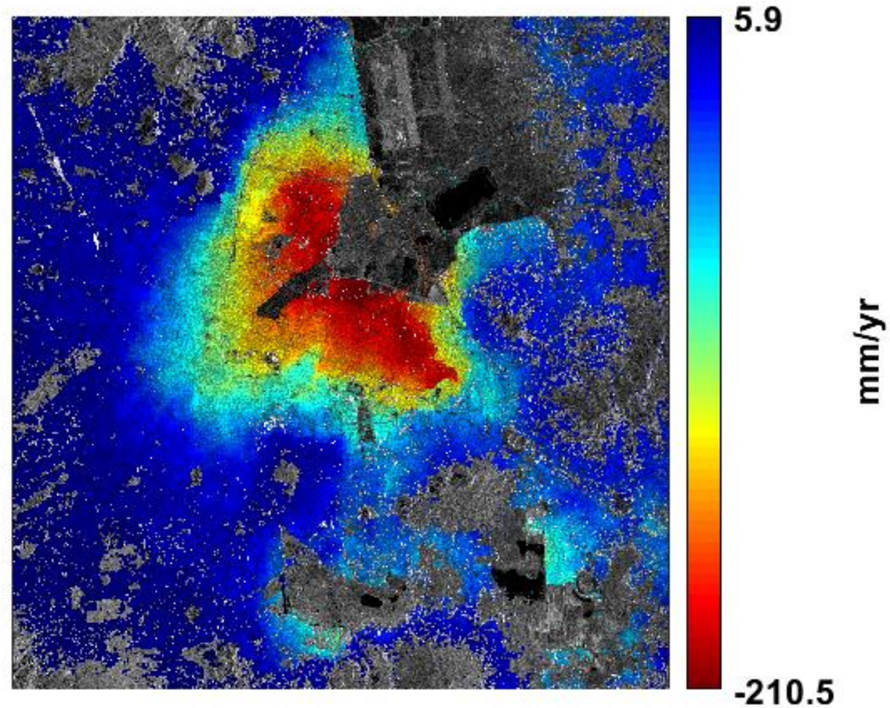
```
>> ps_plot('v')
```

To subtract the DEM error before estimating mean velocity, run

```
>> ps_plot('v-d')
```

The velocity values are relative to the mean velocity of the whole image, by default. You can set a circular reference area using the '*ref_centre_lonlat*' and '*ref_radius*' parameters, or a rectangular reference area using the '*ref_lon*' and '*ref_lat*' parameters. Pick an area in the west of the image and set this as your reference area.

```
>> setparm('ref_centre_lonlat',[-99.19, 19.45])
>> setparm('ref_radius',500)
```



Plotting the time series

>> ps_plot('v-d','ts')

Click on the 'TS plot' button and select an area of the image using the cross hairs.
The time series for the nearest resampled PS pixels will be plotted.

You can set the search radius within the plotted figure. Try changing the radius to 50m and reselecting a point. Now all PS within 50 m of your selected position are plotted, together with the average of these PS.

Try plotting various positions with different standard deviations and experiment with plotting the double difference time series.