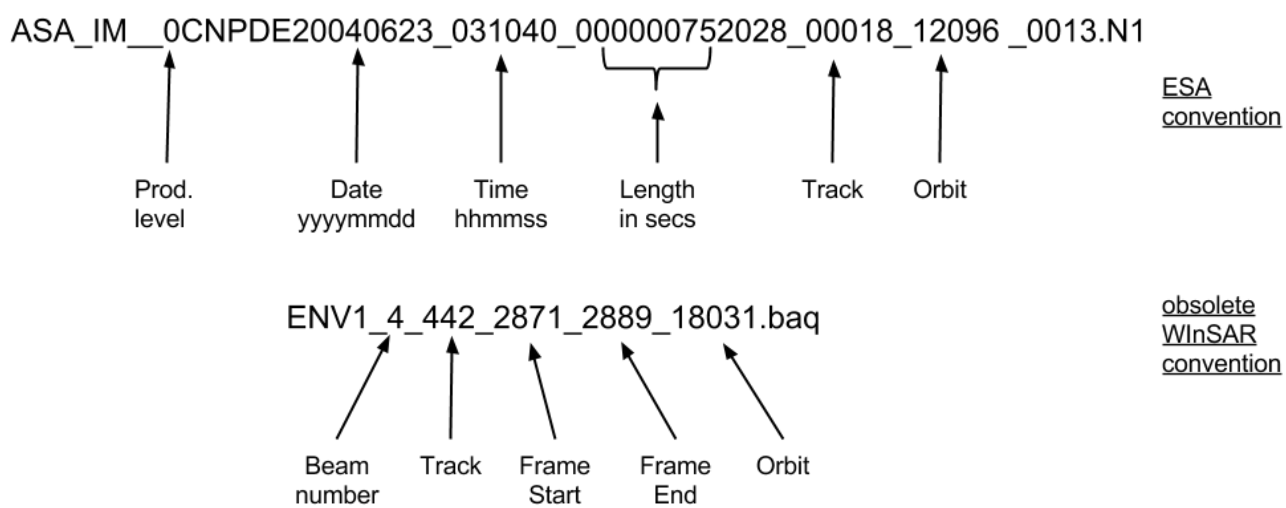
1. Understanding Envisat Data Set Names

In this tutorial we will process an Envisat dataset covering the 2010 M7.2 Baja California, Mexico earthquake (official name El Mayor-Cucapah earthquake).

We will learn how to process SAR data from the European Space Agency's Envisat satellite (actually called advanced SAR or ASAR). Envisat SAR data are typically obtained either directly from ESA, UNAVCO's WInSAR archive or through the GEO Geohazards Supersites archive. The files may be named differently depending on the source.



Some of the files in the WInSAR archive in the past used a different naming convention from the standard ESA naming convention, as shown in the diagram above. All of the Envisat data in the WInSAR archive at UNAVCO is now converted back to the original ESA names, but files downloaded some years ago might have the special ".baq" name. In both cases, the file content is the same, only the name is different.

Envisat SAR data files have both metadata and binary data in the same file. The metadata at the beginning is ASCII text and can be viewed in your terminal, but the binary data later in the file is not directly viewable (and viewing it may confuse your terminal). ISCE presently supports only processing the raw (Level 0) data from Envisat ASAR acquired in imaging or stripmap mode, so make sure the filename starts with ASA_IM__0 if it has the standard ESA name.

2. How to insert Envisat filenames into the ISCE xml input files

The basic ISCE input file for Envisat is similar to that for ALOS PALSAR. If you have not gone through the tutorial Lab 3.1 for running ISCE with ALOS, you should at least read through Step 2 of that tutorial, if not also setting up the input file and processing the ALOS data used in Step 3 of that tutorial. In this tutorial we assume you have read Step 2 of the ALOS tutorial and will only talk about the differences for processing Envisat data.

As in the ALOS and ERS input files the Envisat input files contain the `MASTER` and `SLAVE` component tags that contain the property tags `IMAGEFILE` and `OUTPUT`, but for Envisat there is no `LEADERFILE` tag because the leader information is included in the image file. Different from the ALOS and ERS input files are two new property tags (the ancillary orbit and instrument files) that contain information about the orbit data for the data take and SAR instrument calibration. Both of these properties are required for Envisat data processing.

The way to tell ISCE the orbit and instrument file information is to insert the following `ORBITFILE` and `INSTRUMENTFILE` tags (order is not important) in *both* the `Master` and the `Slave components` (the orbit files will be different for each date and INS files may also differ so we have to specify these for both input scenes):

```
<property name="INSTRUMENTFILE">
     <value>"/home/ubuntu/data/instruments/ENVISAT/ASA_INS_AXVI
     EC20091217_114637_20090428_100000_20101231_235959"</value>
</property>

<property name="ORBITFILE">
     <value>"/home/ubuntu/data/orbits/ENVISAT/VOR/DOR_VOR_AXVF-
     P20100423_084900_20100327_215526_20100329_002326"</value>
</property>
```

The Envisat data files do not contain the orbit information. You have to download the orbit files separately. We will use the DORIS orbit files (DORIS is the name of the method they used to determine the Envisat orbits) that can be downloaded from ESA, which requires registration with ESA. The location of the link to the orbit data (at the moment) is https://earth.esa.int/web/guest/data-access/browse-data-products/-/article/doris-precise-orbit-state-vectors-1502. If this link does not work, then search for “Envisat DORIS orbits” in your favorite search tool. The Delft Institute for Earth-oriented Space Research (DEOS) also provides precise orbits for the early part of the Envisat mission, but they stopped calculating the orbits circa 2007, so they are not so useful. We have installed the necessary Envisat DORIS orbits in the directory `/home/ubuntu/data/orbits/ENVISAT/VOR` The Envisat DORIS orbit file names have three dates and times in their file names, with the first date and time showing the time the file was produced, the second date and time telling the start of the time period covered by the file, and the third date and time telling the end of the period, e.g., `DOR_VOR_AXVF-P20100423_084900_20100327_215526_20100329_002326` was produced 2010/04/23 and covers the time interval from 2010/03/27 21:55:26 to 2010/03/29 00:23:26 or basically the entire day 2010/03/28 plus two hours the day before and 23 minutes the day after. The `DOR` means it is a DORIS orbit file and `VOR` means that it is the final verified orbit. At present, you have to find the correct orbit data file and enter the name in the ISCE input files, unlike the automated search implemented for ERS orbits. Automated search should be added to ISCE later.

You also need the Envisat SAR instrument calibration file or INS file to read the Envisat data. They updated the INS files at several times during the Envisat mission, so there are different files for different time periods. The INS files can be downloaded from ESA's Envisat ASAR auxiliary data directory (http://earth.eo.esa.int/services/auxiliary_data/asar/current/) without registration. The INS file names also have three dates and times in their file names, with the first date and time showing the time the file was produced, the second date and time telling the start of the time period covered by the file, and the third date and time telling the end of the period (just like the orbit file names), e.g., `ASA_INS_AXVIEC20091217_114637_20090428_100000_20101231_235959` was produced 2009/12/17 and covers the time interval from 2009/04/28 to 2010/12/31. Sometimes ESA produced a new INS file before the planned end of a previous file, so you should use the newest file that covers the time interval of the data you have. Again, ISCE does not presently have the capability to search through a directory of INS files and select the correct one for the image files, so you need to manually select the file.

3. Processing Envisat data with ISCE

You should change your directory to the lab5 directory for Envisat processing. We will use the Envisat scenes that we downloaded from UNAVCO earlier for this lab. You can position yourself in that directory from wherever you might be positioned currently as follows:

```
> cd
> cd data/lab5/env
```

There should be two directories there for two different data acquisition dates, if you completed the download earlier:

```
> ls
20100328  20100502
```

Let's check the two data directories to make sure that the files are there. First go into the `20100502` directory:

```
> cd 20100502
> ls
ASA_IM__0CNPDE20100502_175001_000000172089_00084_42723_0354.N1
ASA_IM__0CNPDE20100502_175016_000000172089_00084_42723_0354.N1
```

There should be two data files as shown, because we downloaded two frames for this date. Make sure the files have `20100502` in the name so that the right dates are in this directory. As we saw earlier in this lab, the scenes have `_00084` in the name so they are from Envisat track (relative orbit) 84.

Now let's check the data directory for the other date `20100328`. We can go up and back down into that directory with this `cd` command:

```
> cd ../20100328/
> ls
ASA_IM__0CNPDE20100328_175004_000000162088_00084_42222_9504.N1
ASA_IM__0CNPDE20100328_175019_000000162088_00084_42222_9504.N1
```

Again, check to see if you have the two data files for the `20100328` date as shown. Unlike the ERS data, the Envisat data files do not require any unpacking before we can use them.

To prepare for processing the Envisat data, first make a new processing directory in the `data/lab5/env` directory (first making sure that you are in the proper directory using the `pwd` command or use the `cd` command once without an argument to first move to the top directory and then with the argument `data/lab5/env` to move into the proper directory),

```
> cd
> cd data/lab5/env
> pwd
/home/ubuntu/data/lab5/env

> mkdir 20100502_20100328
> cd 20100502_20100328
> pwd
/home/ubuntu/data/lab5/env/20100502_20100328
```

For Envisat data, you can either use the separate files method, with the input information about each date (master and slave components) in a separate file, or the all-in-one method. Here we will explain the separate file method that is a little easier to read (in my opinion).

First, create the input component file for the 20100502 date:

```
> nano 20100502.xml
```

Then enter the IMAGEFILE, INSTRUMENTFILE, ORBITFILE, and OUTPUT information about this date as described above. For this processing, we will use a list of input images (file names with paths in quotes, inside square brackets) to process the two frames from each date together:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>

<component name="Envi">
    <property name="IMAGEFILE">

<value>['../20100502/ASA_IM__0CNPDE20100502_175001_000000172089_00084
_42723_0354.N1','../20100502/ASA_IM__0CNPDE20100502_175016_0000001720
89_00084_42723_0354.N1']</value> <!-- image files -->
    </property>
    <property name="INSTRUMENTFILE">

<value>"/home/ubuntu/data/instruments/ENVISAT/ASA_INS_AXVIEC20091217_
114637_20090428_100000_20101231_235959"</value> <!-- instrument file
-->
    </property>
    <property name="ORBITFILE">

<value>"/home/ubuntu/data/orbits/ENVISAT/VOR/DOR_VOR_AXVF-P20100604_1
04000_20100501_215526_20100503_002326"</value> <!-- orbitfile -->
```

```
    </property>
    <property name="OUTPUT">
        <value>"20100502.raw"</value> <!-- output raw file -->
    </property>
</component>
```

Let's check that the paths for some of the files specified in the XML work on your system (watch out for extra breaks in the long lines):

```
> ls
../20100502/ASA_IM__0CNPDE20100502_175001_000000172089_00084_42723_03
54.N1
../20100502/ASA_IM__0CNPDE20100502_175001_000000172089_00084_42723_03
54.N1
> ls
/home/ubuntu/data/instruments/ENVISAT/ASA_INS_AXVIEC20091217_114637_2
0090428_100000_20101231_235959
/home/ubuntu/data/instruments/ENVISAT/ASA_INS_AXVIEC20091217_114637_2
0090428_100000_20101231_235959
```

Now create the input component file for the `20100328` date, and again enter all the required information for this component:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>

<component name="Envi">
    <property name="IMAGEFILE">

<value>['../20100328/ASA_IM__0CNPDE20100328_175004_000000162088_00084
_42222_9504.N1','../201
00328/ASA_IM__0CNPDE20100328_175019_000000162088_00084_42222_9504.N1'
]</value> <!-- image files -->
    </property>
    <property name="INSTRUMENTFILE">

<value>"/home/ubuntu/data/instruments/ENVISAT/ASA_INS_AXVIEC20091217_
114637_20090428_100000_
20101231_235959"</value> <!-- instrument file -->
    </property>
    <property name="ORBITFILE">

<value>"/home/ubuntu/data/orbits/ENVISAT/VOR/DOR_VOR_AXVF-P20100423_0
84900_20100327_215526_2
```

```
0100329_002326"</value> <!-- orbitfile -->
    </property>
    <property name="OUTPUT">
        <value>"20100328.raw"</value> <!-- output raw file -->
    </property>
</component>
```

Now we need to create the main `insarApp.xml` file. We will use the `20100502` scene as the master scene, and we will add some other properties here. The first one is to set the `posting` or spacing of pixels in the interferogram, so that insarApp will adjust the number of looks or SLC pixels averaged in the interferogram. We will also request phase unwrapping and use the “`icu`” unwrapping program.

```
<?xml version="1.0" encoding="UTF-8"?>
<insarApp>
<component name="insar">
<!-- Posting is automatically calculated if not specified here.
-->
        <property name="Posting">
            <value>40</value>
        </property>
    <property name="unwrap">
        <value>True</value>
    </property>
        <property name="unwrapper name">
            <value>icu</value>
        </property>
        <property name="Sensor Name">
            <value>Envisat</value>
        </property>
        <property name="Doppler Method">
            <value>useDOPIQ</value>
        </property>
        <component name="Master">
            <catalog>20100502.xml</catalog>
        </component>
        <component name="Slave">
            <catalog>20100328.xml</catalog>
        </component>
</component>
</insarApp>
```

Now we are ready to run the processing, using the steps option, in case we want to re-run steps later. You may want to take a coffee or water break.

```
> insarApp.py insarApp.xml --steps
```

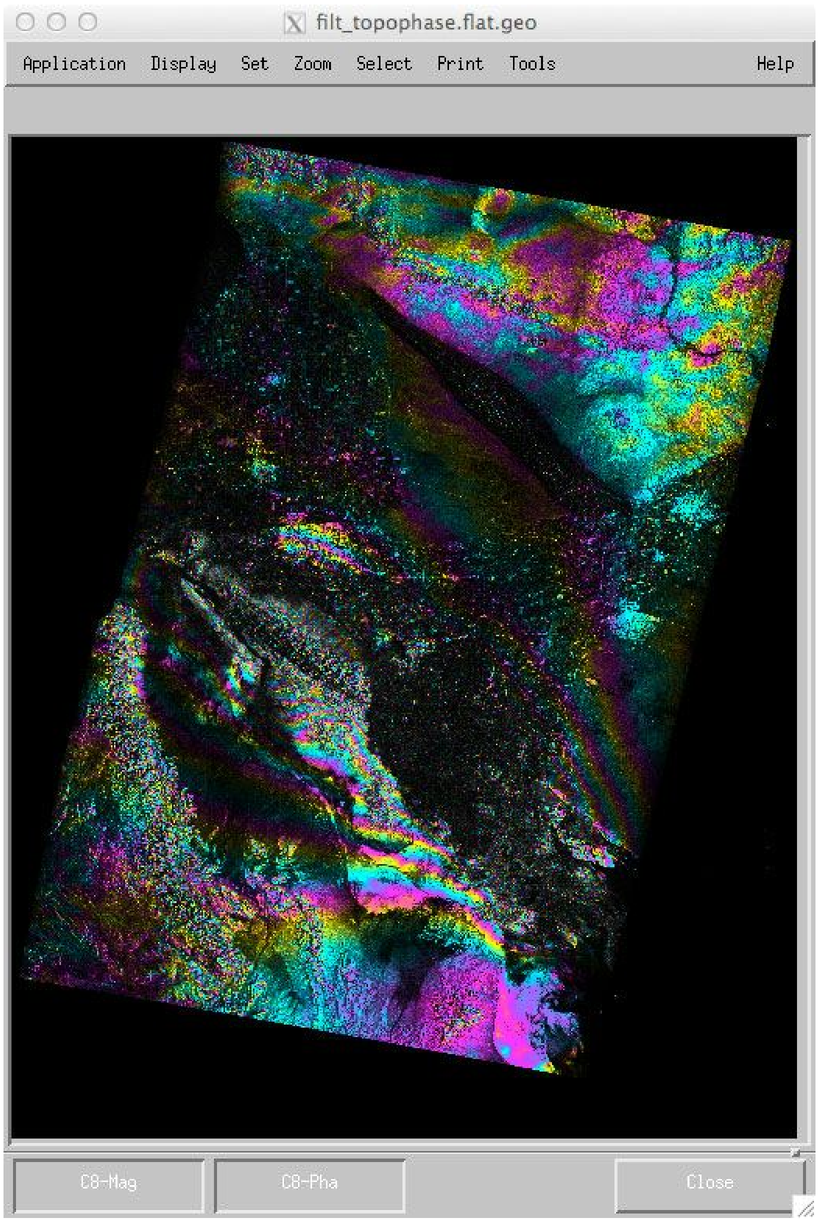
4. Your completed run

At the end of the processing, you should see something like this:

```
########################################################################
################################
2014-07-30 17:29:18,420 - isce.insar.runGeocode - INFO -
########################################################################
################################
    runGeocode - Outputs
------------------------------------------------------------------------
--------------------------------
runGeocode.outputs.LONGITUDE_SPACING = 0.0008333333333333334
runGeocode.outputs.MINIMUM_GEO_LONGITUDE = -116.09083333333334
runGeocode.outputs.LATITUDE_SPACING = -0.0008333333333333334
runGeocode.outputs.MAXIMUM_GEO_LATITUDE = 31.355000000000004
runGeocode.outputs.MAXIMUM_GEO_LONGITUDE = -114.4175
runGeocode.outputs.GEO_LENGTH = 2560
runGeocode.outputs.MINIMUM_GEO_LATITUDE = 33.487500000000004
runGeocode.outputs.GEO_WIDTH = 2009
########################################################################
################################
2014-07-30 17:29:18,420 - isce.insar - INFO - Total Time: 807 seconds
```

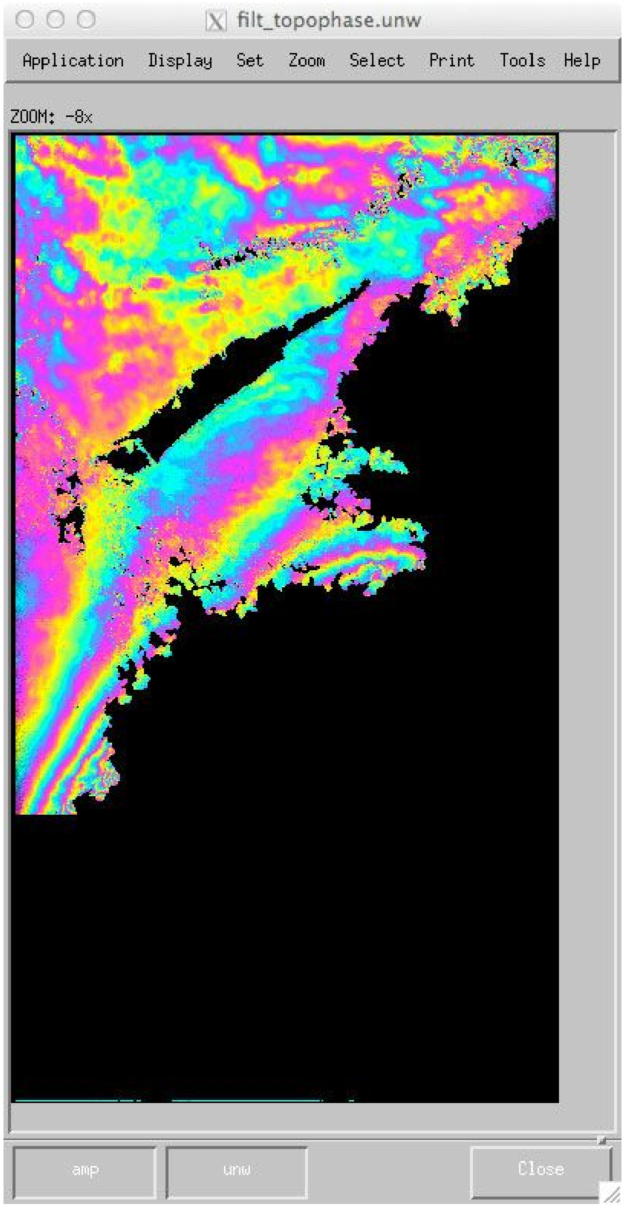
Note that ISCE has downloaded the SRTM 3-arcsecond DEMs, and the final geocoding has this spacing (`LONGITUDE_SPACING = 0.0008333333333333334`), because part of the area is outside the USA so the 1-arcsecond SRTM data is not publicly available (yet). Let's have a look at the geocoded wrapped interferogram (it is large, so we tell MDX to zoom out by 4 to start):

```
> mdx.py filt_topophase.flat.geo -z -4 &
```



Now we can see a lot of fringes! This is the deformation from the M7.2 earthquake. Let’s look at the unwrapped phase now, in radar coordinates:

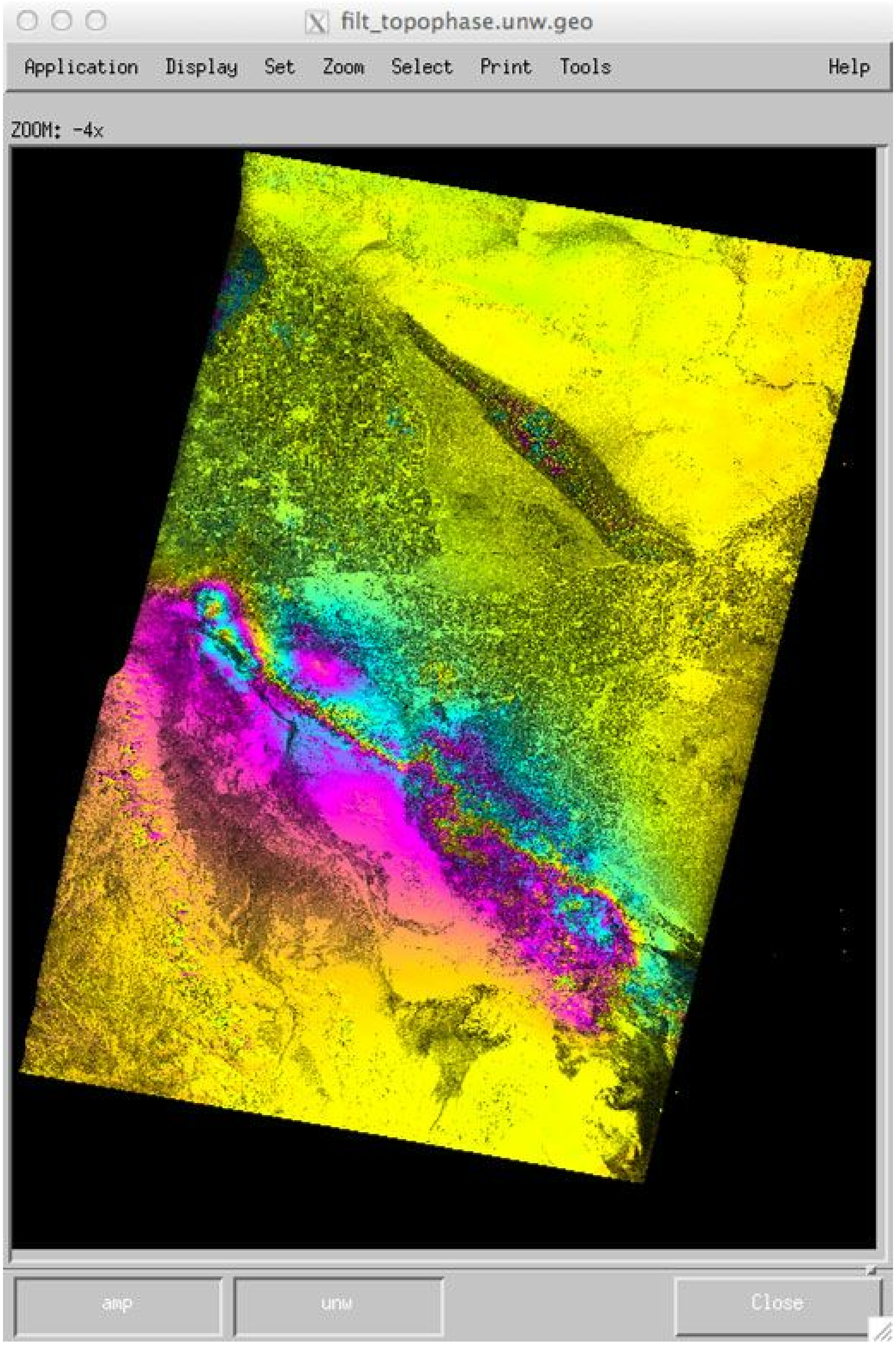
```
> mdx.py filt_topophase.unw -z -8 &
```



We can see that the “icu” unwrapper only managed to unwrap the top (north) side of the fault rupture, and it stopped unwrapping after the first patch (3700 lines of the multi-look interferogram). Let’s try a different unwrapping program called SNAPHU (Statistical-cost, Network-flow Algorithm for Phase Unwrapping) written by Curtis Chen when he was at Stanford (see Chen and Zebker, 2002). Edit your insarApp.xml file and change the “unwrapper name” property value from “`icu`” to “`snaphu`”. Since we ran the initial processing with “`--steps`”, we can just restart the processing at the `unwrap` step without having to rerun the earlier steps (we also don’t need to specify the `--steps` flag in addition to the `--start=unwrap`):

```
> insarApp.py insarApp.xml --start=unwrap
```

The SNAPHU program can take a while to run, depending on how large and noisy your interferogram is. Low coherence, noisy data take a long time to unwrap and “snaphu” unwraps everything, unlike the “grass” and “icu” unwrappers that mask out the noisy areas before unwrapping. When the unwrapping is complete, take a look at the `filt_topophase.unw` or `filt_topophase.unw.geo` file with `mdx.py`. Change the color wrap on the phase to 100 radians to see the large displacement from the earthquake like this (I also adjusted the exponent of the amplitude image to 0.5):



The 2010 M7.2 earthquake ruptured about 120 km of faults in northern Mexico. Rocks on the southwest side of the fault moved up to 2 meters northwest relative to the northeast side. See this paper for more information:

Wei, S., Fielding, E. J., Leprince, S., Sladen, A., Avouac, J.-P., Helmberger, D. V., Hauksson, E., Chu, R., Simons, M., Hudnut, K. W., Herring, T., & Briggs, R. W. (2011). Superficial simplicity of the 2010 El Mayor–Cucapah earthquake of Baja California in Mexico. *Nature Geosci, 4*, 615-618. http://www.nature.com/ngeo/journal/v4/n9/full/ngeo1213.html