1. Introduction to basic ISCE

Welcome to your first session using ISCE. The purpose of this session is to demonstrate the most straightforward application of ISCE for geodetic imaging, the insarApp.py application. insarApp takes two raw data files and optionally a digital elevation model (DEM) and produces a geocoded, and optionally unwrapped, interferogram. insarApp runs automatically from start to finish with no required interaction. The input files have a number of configurable parameters that allow the user to control how the processing is done. For this first session, we have preconfigured all the input parameters in a set of input files.

So let’s get started and simply explore what and where the data files are, then run insarApp.

2. Where are we?

Let's see what the files look like on the disk. First let's see where we are. The Unix "pwd" command tells us the "present working directory" in the file system.

```
> pwd
/home/ubuntu/
```

"ubuntu" is your predefined username on this cloud instance virtual machine. (Ubuntu happens to be the name of a version of the linux operating system that is running on this virtual machine.) We have chosen a single username for every instance so everyone sees the same interface.

Now let's take a look at what we've got. The unix "ls" command lists files in the present working directory

```
> cd data
> ls
giant       lab1  lab4      orbits                    sites
instruments  lab3  lost+found  raw_from_roi_pac_tutorial
```

This shows that within the present working directory are a number of other directories that contain the information and descriptive content for each of the ISCE labs. Let's change our working directory to lab1 so we can get started with the first lab:

```
> cd lab1
> pwd
/home/ubuntu/data/lab1
```

3. Setting up to run ISCE.

Now let’s see what is in this directory using the “ls -l” command to display more (“-l” = “longer”) information about the files:

```
> ls -l
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 18 17:33 20061231
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 18 17:34 20070215
drwxrwxr-x 3 ubuntu ubuntu 4096 Oct 17 05:48 20070215_20061231
```

Note there are three subdirectories within the lab1 directory: 20061231 contains the raw data for one radar observation, 20070215 contains the raw data for the other radar observation, and 20070215_20061231 is a directory we have created that will be used to store the output of the insarApp run. This convention of naming data directories by the date is common in the InSAR community. You probably know the dates of your acquisitions from when you decided what data to process. We will describe later how to view the metadata to determine the date of any data set. We can see what is in these subdirectories while staying at this level by issuing the following command:

```
> ls 20061231
IMG-HH-ALPSRP049770670-H1.0__A  LED-ALPSRP049770670-H1.0__A
> ls 20070215
IMG-HH-ALPSRP056480670-H1.0__A  LED-ALPSRP056480670-H1.0__A
> ls 20070215_20061231
insar_20070215_20061231.xml  Master.xml  Slave.xml
```

The data set we have provided for this session is a pair of ALOS PALSAR observations acquired over Los Angeles. We have named the directories containing these data by using the date of acquisition, but the user is free to name these directories arbitrarily (within unix requirements). In this example, we have provided an appropriate DEM that will be applied in the processing. In a later step, we will show how to tell ISCE to find an appropriate DEM, and configure parameters.

To run insarApp, we need to set the present working directory to the directory where we want the output to be. In addition, the input configuration XML files must be in this same working directory. From above, we see that the xml file is in 20070215_20061231, so let’s change to that directory:

```
> pwd
> cd 20070215_20061231
> pwd
/home/ubuntu/data/lab1/20070215_20061231
```

```
> ls
insar_20070215_20061231.xml  Master.xml  Slave.xml
```

We will come back later to examine the contents of the xml file, but let’s jump in the deep end and process some data from start to finish.

4. Running ISCE and examining the output

Now that we are at the appropriate working directory to run the application, to process the data we simply issue the following command:

```
> insarApp.py insar_20070215_20061231.xml
```

At this point, the screen should fill with scrolling text describing where you are in the processing. For the scenes provided, the total processing time should take roughly 20 minutes. Go get some coffee and come back then.

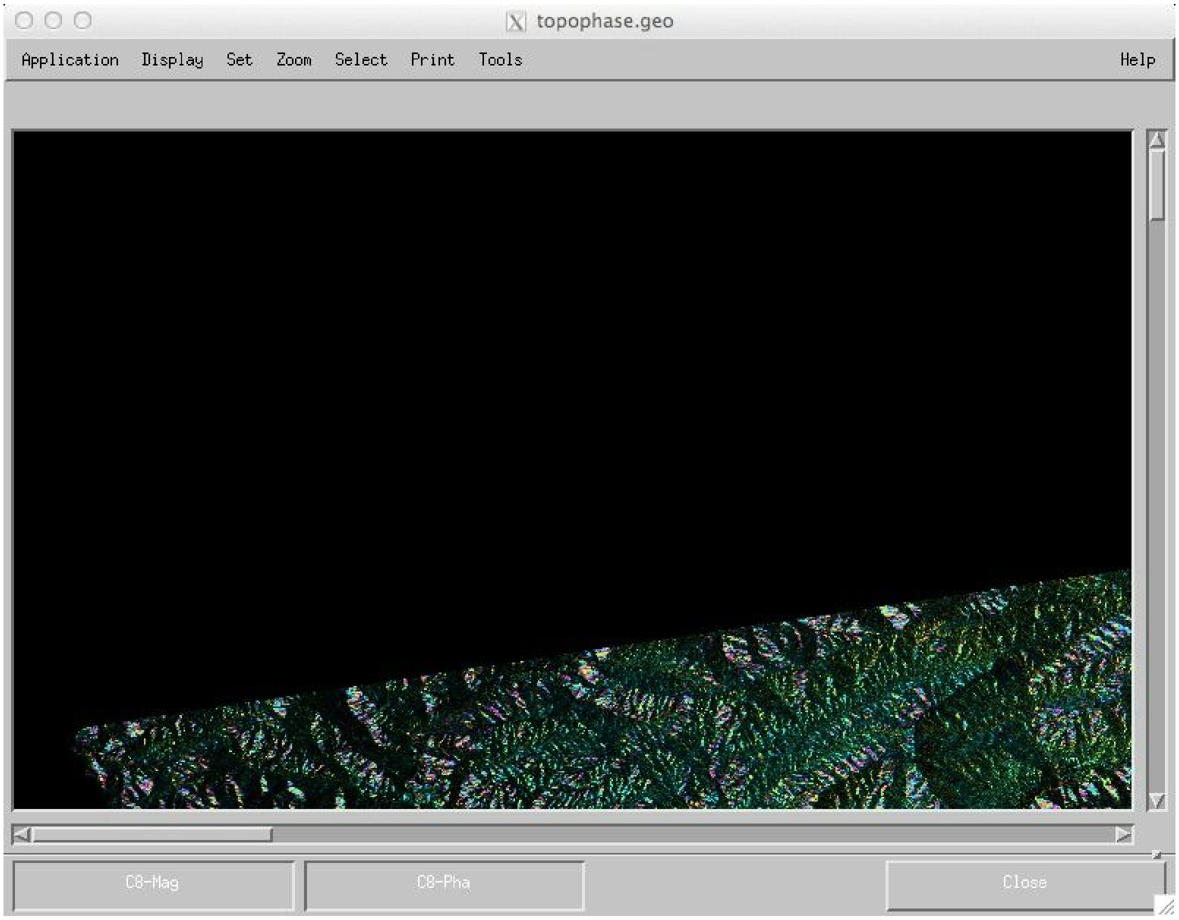
Now that insarApp has finished, we can examine the output, and try to make sense of some of the ancillary data that ISCE computes. First, let's see what we've got:

```
> ls
20061231.raw                                  iz
20061231.raw.aux                              lat
20061231.raw.xml                              lon
20061231.slc                                  Master.xml
20061231.slc.xml                              rangeOffset.mht
20070215.raw                                  rangeOffset.mht.xml
20070215.raw.aux                              resampImage.amp
20070215.raw.xml                              resampImage.amp.xml
20070215.slc                                  resampImage.int
20070215.slc.xml                              resampImage.int.xml
azimuthOffset.mht                             resampOnlyImage.amp
azimuthOffset.mht.xml                         resampOnlyImage.int
catalog                                       resampOnlyImage.int.xml
dem.crop                                      rgdem
demLat_N33_N35_Lon_W119_W116.dem              Slave.xml
demLat_N33_N35_Lon_W119_W116.dem.wgs84        topophase.cor
demLat_N33_N35_Lon_W119_W116.dem.wgs84.xml     topophase.cor.xml
demLat_N33_N35_Lon_W119_W116.dem.xml          topophase.flat
filt_topophase.cor                            topophase.flat.xml
filt_topophase.flat                           topophase.geo
filt_topophase.flat.xml                       topophase.geo.xml
filt_topophase.unw                            topophase.mph
insar_20070215_20061231.xml                   topophase.mph.xml
insar.log                                     z
insarProc.xml                                 zsch
isce.log
```

In later labs we will examine the most important files in this directory in detail, but for now, let's do the most gratifying thing and just look at the final output, which is called “filt_topophase.flat.geo.” To display InSAR data, we have developed a tool called mdx that is especially well tailored to examining large files with complex number data types such as interferograms. We have an application that looks at the metadata associated with a data file and calling mdx appropriately.

```
> mdx.py filt_topophase.flat.geo
```

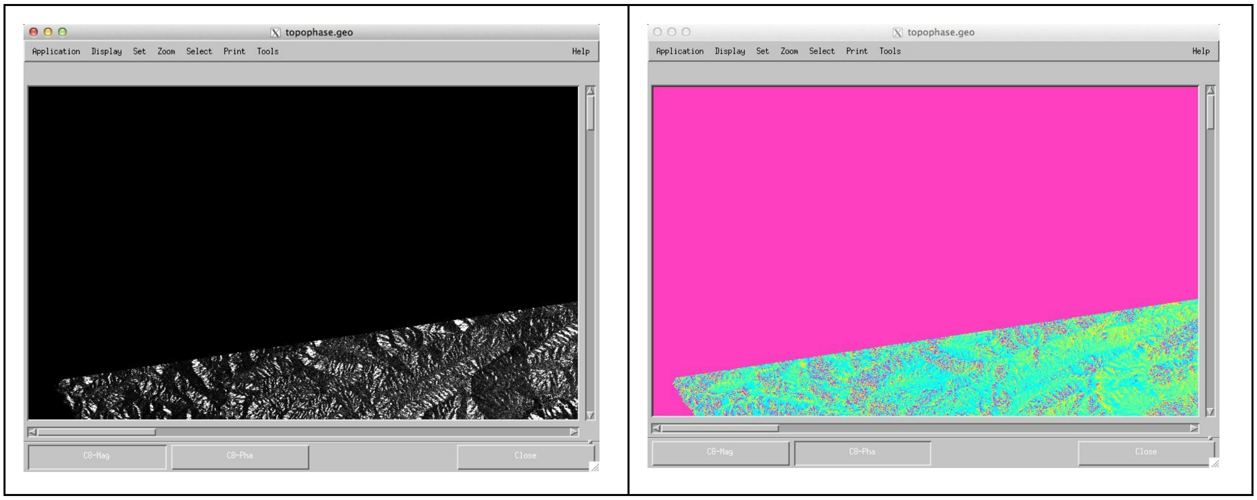
We now have on our screen a new window displaying a portion of the image file.



The data file filt_`topophase.flat.geo` contains a wrapped interferogram that has been flattened and geocoded using a dem (digital elevation model) that was downloaded by ISCE early in the processing. Each pixel in the image is a complex number with an amplitude and phase associated with it. The displayed image has a color mapping that blends the amplitude of the image with the phase which aids the viewer in interpreting the phase values in the context of the local area (e.g. water is always decorrelated in differential InSAR images, so the phase will be random. observing a water body in the amplitude data will assure the viewer that all is well.)

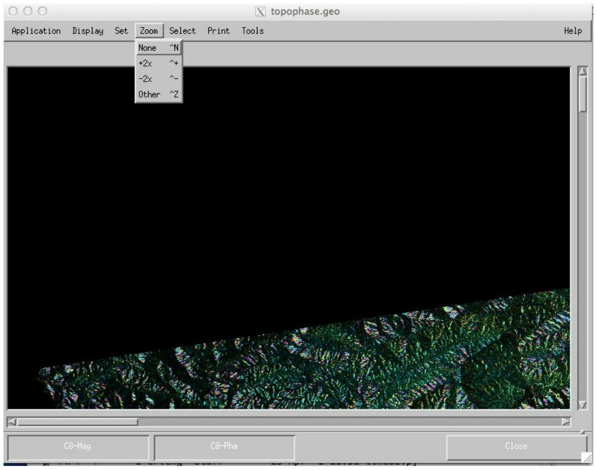
5. Basic MDX controls

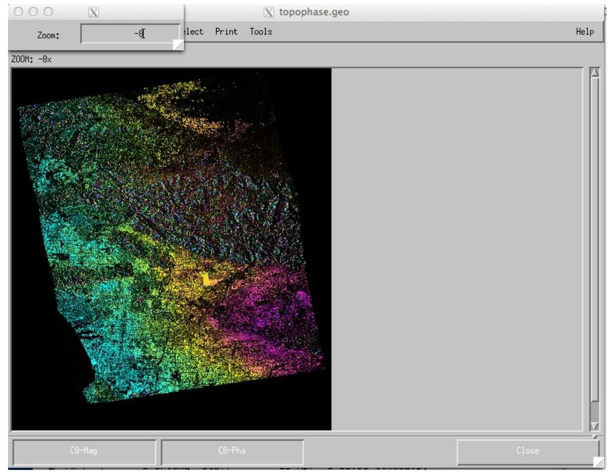
The user can toggle the display to see only amplitude by clicking on the “C8-mag” button. To see only phase, click on “C8-pha”. To go back to seeing both, center click again on “C8-mag”.



mdx has a rich set of display features that can be explored at leisure. We should look at one of the most important features though, which is “zoom”. We see that the image as displayed is only a portion of the full image, displayed at a 1:1 zoom scale. We can scroll around the image by clicking the scroll bars, but to get the big picture, we can select the zoom menu item to zoom out.

The zoom menu presents us with several options. “None” means revert to 1:1 scaling. “+2” means zoom in by a factor of 2. “-2” means zoom out by a factor of 2. “Other” allows us to set the zoom factor as we like. In the images below, we selected a factor of -8 which shows us the entire file. Due to the aspect ratio of the window, extra area is displayed as gray. We can resize the windows to make better use of the screen space.





This concludes your first successful ISCE run!