These notes explain the satellite archive, explain what images are there, and provide examples of matlab sessions to explain how to use both the gridded and non-gridded satellite data that, in addition to PNG format images, are stored there in compact form. The next few paragraphs are descriptive and may be hard to follow if you aren't flipping through the archive. After that, there are examples that are easiest to understand if you go through them from within Matlab.

Data & images are stored for each source file (i.e. distinct .nc or .hdf format data files obtained by ftp) in each of the consecutively numbered directories ('T0', 'T1', …) within the archive structure at latmix.apl.washington.edu:files/INFLO/ .

There are two logins to latmix, one with read-only permission (sftp latmixro@latmix.apl.washington.edu) and one with read-write-permission (sftp latmixrw@latmix.apl.washington.edu) that will also support ssh, scp & rsync. Please use the read-only login if that's all you will be doing.

The directory structure is meant to be interactive, so that you may upload from the field and begin receiving tailor-made images that are modified in geographical or measured value scope from the default target 'T0' choices in the directory files/INFLO/T0. Before attempting to do this please explore this document and the archive as latmixro, and communicate with me (Harcourt@apl.washington.edu ; 206 221 4662) before trying it out.

If you don't recall the passwords please request them individually. This file and any subsequent revisions will be placed in files/INFLO/.

Each relevant image is numbered as they are obtained. This is not necessarily their chronological measurement sequence. Images and data are filed in a directory bearing the UTC reference date of the image (e.g. 15-May-2007). Each different satellite data source gets its own directory containing the dated subfolders. Each targeted region of interest gets a full complement of subfolders for each satellite data source.

To keep individual files small or focus on different geographical areas or measurement scales, smaller or different targets named 'T#' may be generated, where the default target it T0 and a first target T1 is already in place for AVHRR SST images. Each of these sub-domain "targets" gets its own directory, eg. files/INFLO/T1 for T1.

The satellite database contains image and data files for both SST and for its gradient magnitude, and (new this time) vector gradient in corresponding files with 'Grad' and 'Dgrad' inserted before 'SST' in the file names. The gradient maps allow you to gauge the relative strength of fronts, which can be difficult to diagnose consistently from color-mapped SST. The vector Gradient maps show direction in color and magnitude as intensity (black is small or 0). Another issue is that the higher-resolution SST data are accompanied by cloud masks that are variable and imperfect, particularly as they associate strong fronts with cloud cover. The gradient images have the cloud mask 'peeled back', while the SST images do not for the MODIS images, and have a low threshold (25% cloud) for the AVHRR. The AVHRR threshold currently may be adjusted for targets with that sensor.

When more than one satellite pass contributes to the same date in a target domain, a composite can be constructed. These automatically-generated composites are not always satisfying, and are currently produced for AMSRE, TMI and MODIS

images. These images are located in the dated directory, above the individual contribution directories. These files have “Comp” in the name. Composites can be helpful in identifying how much data from a satellite there is on a given day, but they can be confused, especially when they combine partial day and night images. Looking at the size of the most recent SST composite may be an indication of how much data is available for that day. This is best determined by using multiple connections and parsing the addresses manually. Otherwise with a slow connection you will find clicking back and forth frustrating. Identifying which pass contributed the most interesting part of the composite is not always unambiguous, but the size of the compressed data files discussed next may be an indicator.

Data is placed in this archive system in two forms. One form is a matlab file structure compressed using bzip2 (don’t leave home without it), and the other is a more compact binary form. Both are constructed from within Matlab (v. 2010a), and so my directions on how to get them and load the data are too. If you think you might use these files, please check before leaving home that your version of matlab is sufficiently up to dat for reading these files.

```
%...Here are some examples of loading. They are based by previous iteration of
%...this archive, but should help navigating the data files until edited for
%...the latmix data.
%...>> indicates matlab session; %... are my commentary.

%...The matlab and binary data files are compressed using bzip2/bunzip2.
%...Seems everybody has this, but make sure that you do.

>> !bunzip2 AMSRE1data.mat.bz2
>> load AMSRE1data.mat
>> whos
  Name          Size           Bytes  Class

  sstdata       1x1            29808  struct array

Grand total is 20062 elements using 29808 bytes

>> sstdata
sstdata =
    source: [1x1 struct]
      time: [1x1 struct]
      lonE: [1x1 struct]
       lat: [1x1 struct]
       sst: [1x1 struct]
>> sstdata.source
ans =
    ftpnm: [1x98 char]
%...The ftp address of the data source. With bandwidth, you download it
%...and load it into matlab using the netcdf utilities if you want to see the
%...data in higher precision:
>> sstdata.source.ftpnm
ans =
ftp://ftp.misst.org/amsre/swath/nc/2007/20070515-AMSRE-REMSS-L2P-
amsr_l2b_v05_r26755.dat-v01.nc.gz

>> sstdata.time
ans =
    tvec: [2007 5 15 10 29 9.0000]
%...This is the UTC date (i.e. 'datevec' in matlab) associated with the image.
%...I think that is the time of the initial point in the source file, of which
%...the data in this target will be a small subset.
%...This tvec says "15 May 2007 10:29:09Z"

>> sstdata.sst
ans =
      min: 20.8300
      max: 23.7800
    scale: 4.5014e-05
    bytes: [85x77 uint16]
     make: [1x94 char]
>> sstdata.sst.make
ans =
sst=double(sstdata.sst.bytes);sst(sst==0)=NaN;sst=sstdata.sst.min+(sst-
0.5)*sstdata.sst.scale;
>> eval(sstdata.sst.make)
>> whos
  Name          Size           Bytes  Class
```

```
  ans           1x94             188  char array
  sst          85x77           52360  double array
  sstdata       1x1            29808  struct array

Grand total is 26701 elements using 82356 bytes

>> sstdata.lonE
ans =
      min: 207.0000
      max: 213.0000
    scale: 0.0235
    bytes: [85x77 uint8]
     make: [1x103 char]
>> sstdata.lat
ans =
      min: 25
      max: 29
    scale: 0.0157
    bytes: [85x77 uint8]
     make: [1x95 char]
>> eval(sstdata.lonE.make)
>> eval(sstdata.lat.make)
>> figure
>> pcolor(lonE-360,latN,sst),shading interp, colorbar
??? Undefined function or variable 'latN'.

>> pcolor(lonE-360,lat,sst),shading interp, colorbar
>> cd ../../M
No completions found.
>> cd ../../../MODIS_T/15-May-2007/MODIS_T1/
>> clear all global
>> close all
>> ls
ans =
GradMODIS_T1.jpg  gridat  MODIS_T1data.mat.bz2  MODIS_T1.jpg

>> !bzip2 ../../../AMSRE/AM
No completions found.
>> !bzip2 ../../../AMSRE/
15-May-2007/  16-May-2007/  readtarget.m
>> !bzip2 ../../../AMSRE/15-May-2007/AMSRE1/AMSRE1data.mat
>> !bunzip2 MODIS_T1data.mat.bz2
>> load MODIS_T1data
>> whos
  Name          Size           Bytes  Class

  ans           1x60             120  char array
  sstdata       1x1            99118  struct array

Grand total is 72046 elements using 99238 bytes

>> eval(sstdata.sst.make)
>> whos
  Name          Size           Bytes  Class

  ans           1x60             120  char array
```

```
  isgood      486x49          190512  double array (logical)
  sstdata       1x1            99118  struct array
  sstfull     486x49          190512  double array
  sstgood     486x49          190512  double array

Grand total is 143488 elements using 670774 bytes

>> eval(sstdata.lonE.make)
>> eval(sstdata.lat.make)
>> figure
>> pcolor(lonE-360,lat,isgood),shading interp, colorbar
>> pcolor(lonE-360,lat,sstfull),shading interp, colorbar
>> pcolor(lonE-360,lat,sstgood),shading interp, colorbar
```