Reading and Visualizing NetCDF data from GOES

Matlab contains functions that makes it easy to read and visualize geospatial data. In this script we will look at two functions that can be used to interact with NetCDF's and ways we can visualize the data in them.

NetCDF Functions

For the GOES data, there will always at least be 2 NetCDF files that you must work with. The first one is the file with the Latitude and Longitude coordinates, and the second one is the NetCDF with the LST data.

First we will use a function for reading any NetCDF file. Here 'ncread' is given the filename and the variable of interest. Here we load the Latitude and Longitude into our MATLAB workspace.

```
Lat = ncread('GOESR_ABI_CONUS_East.nc', 'Latitude');
Lon = ncread('GOESR_ABI_CONUS_East.nc', 'Longitude');
```

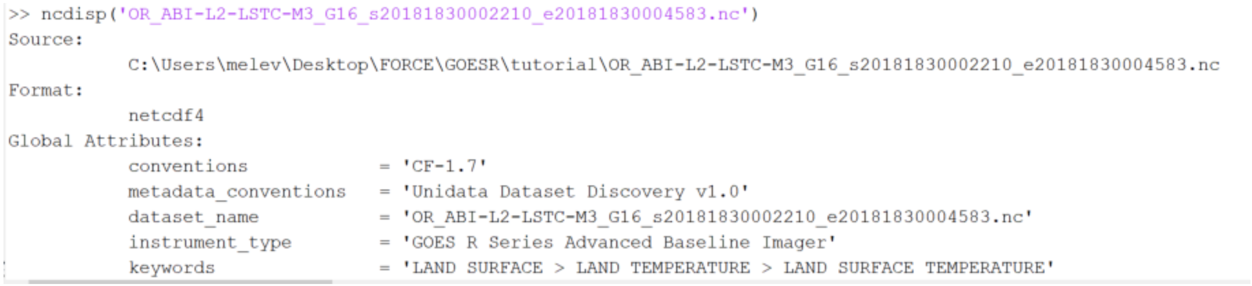
We will also use the same function to read the GOES Land Surface Temperature (LST) data.

```
GOES_LST = ncread('OR_ABI-L2-LSTC-M3_G16_s20181830002210_e20181830004583.nc','LST');
```

The function 'ncdisp' can be used to look at the metadata for the GOES data. This may be useful if you are interested in the particular 'Attributes' like units, the full name of a variable, how the variable handles missing data, etc.

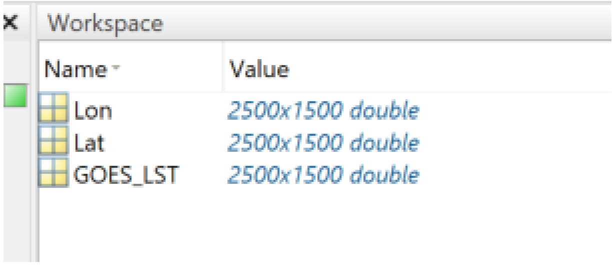
```
ncdisp('OR_ABI-L2-LSTC-M3_G16_s20181830002210_e20181830004583.nc');
```

The call to ncdisp has been suppressed because a lot of information is displayed. Below is image of the first few lines of output of the call to ncdisp.



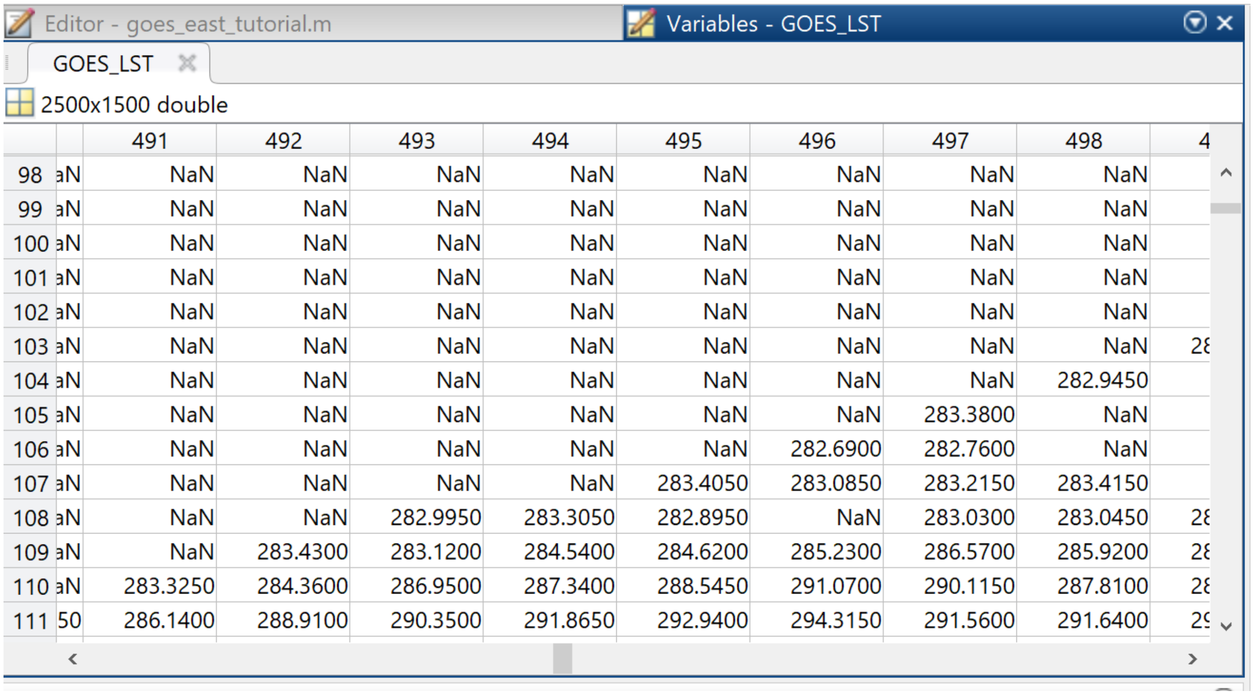
```
>> ncdisp('OR_ABI-L2-LSTC-M3_G16_s20181830002210_e20181830004583.nc')
Source:
           C:\Users\melev\Desktop\FORCE\GOESR\tutorial\OR_ABI-L2-LSTC-M3_G16_s20181830002210_e20181830004583.nc
Format:
           netcdf4
Global Attributes:
           conventions              = 'CF-1.7'
           metadata_conventions     = 'Unidata Dataset Discovery v1.0'
           dataset_name             = 'OR_ABI-L2-LSTC-M3_G16_s20181830002210_e20181830004583.nc'
           instrument_type          = 'GOES R Series Advanced Baseline Imager'
           keywords                 = 'LAND SURFACE > LAND TEMPERATURE > LAND SURFACE TEMPERATURE'
```

If you take a look at your workspace you will notice that all 3 variables are the same size.



Name	Value
Lon	2500x1500 double
Lat	2500x1500 double
GOES_LST	2500x1500 double

Double-clicking on these variables will show you the contents of each variable. Below is image of the GOES LST data. Here I have scrolled to a region in the matrix where there is both NaN ("Not a number", ie. missing data) and LST values.



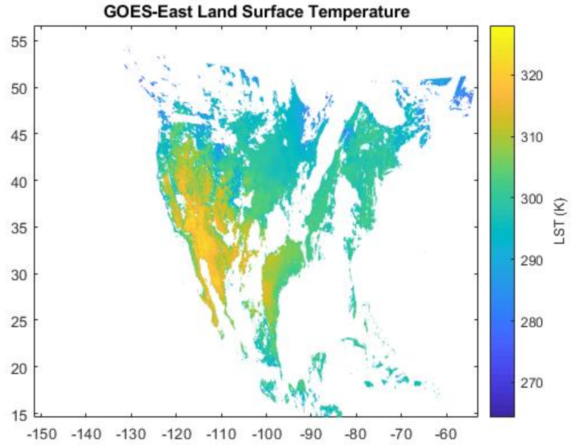
		491	492	493	494	495	496	497	498	4
98	aN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
99	aN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
100	aN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
101	aN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
102	aN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
103	aN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	28
104	aN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	282.9450	
105	aN	NaN	NaN	NaN	NaN	NaN	NaN	283.3800	NaN	
106	aN	NaN	NaN	NaN	NaN	NaN	282.6900	282.7600	NaN	
107	aN	NaN	NaN	NaN	NaN	283.4050	283.0850	283.2150	283.4150	
108	aN	NaN	NaN	282.9950	283.3050	282.8950	NaN	283.0300	283.0450	28
109	aN	NaN	283.4300	283.1200	284.5400	284.6200	285.2300	286.5700	285.9200	28
110	aN	283.3250	284.3600	286.9500	287.3400	288.5450	291.0700	290.1150	287.8100	28
111	50	286.1400	288.9100	290.3500	291.8650	292.9400	294.3150	291.5600	291.6400	29

Visualizing GOES LST Data

Here we show you two simple methods for quickly visualizing GOES LST data. One method uses the default MATLAB function and the other uses the Mapping Toolbox. Most of the names of functions in the Mapping Toolbox are have the same name as the functions from the default MATLAB with an 'm' appended to it.

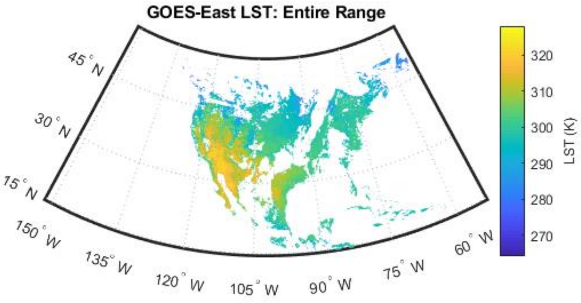
Here we will use the 'pcolor' function to visualize the data. This creates a checkerboard-like image for the data.

```
figure(1)
pcolor(Lon, Lat, GOES_LST)
shading flat
h = colorbar; set(get(h, 'label'), 'string', 'LST (K)') % Add the axis for the LST data
title("GOES-East Land Surface Temperature")
```



Using the MATLAB Mapping toolbox we can use aadditional functions that can interpret the spatial coordinates more readily.

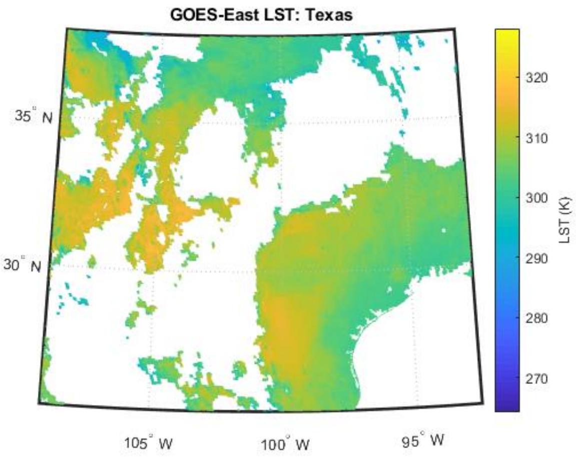
```
figure(2)
worldmap([min(min(Lat)) max(max(Lat))], [min(min(Lon)) max(max(Lon))])
pcolorm(Lat, Lon, GOES_LST) % different order than the default 'pcolor'
h = colorbar; set(get(h, 'label'), 'string', 'LST (K)')
title("GOES-East LST: Entire Range")
```



The 'worldmap' function sets the limits of the bounding box that will encompass the region we are looking at. The inputs used here are the minimum and maximum for the latitude and the longitude, respectively.

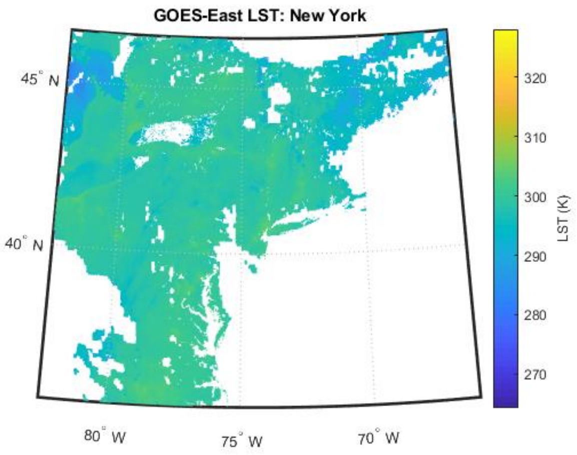
You can specify the two extreme corners for your region of interest. Below is the code to visualize the same data as before but only for Texas.

```
figure(3)
worldmap([25.32, 37.93], [-109.06, -92.6])
pcolorm(Lat, Lon, GOES_LST)
h = colorbar; set(get(h, 'label'), 'string', 'LST (K)')
title("GOES-East LST: Texas")
```



We can do the same thing for New York.

```
figure(4)
worldmap([35.35, 46.53], [-82.51, -66.06])
pcolorm(Lat, Lon, GOES_LST)
h = colorbar; set(get(h, 'label'), 'string', 'LST (K)')
title("GOES-East LST: New York")
```



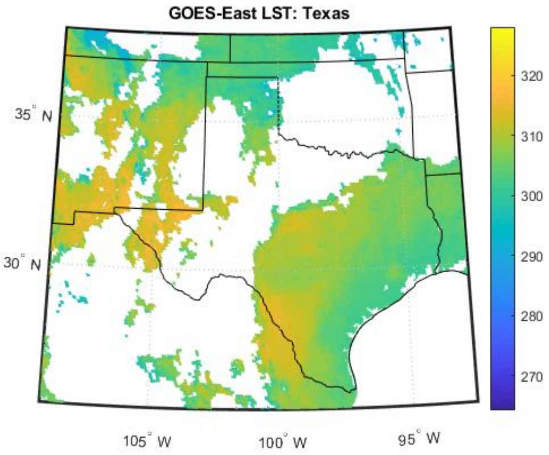
Adding A Shapefile

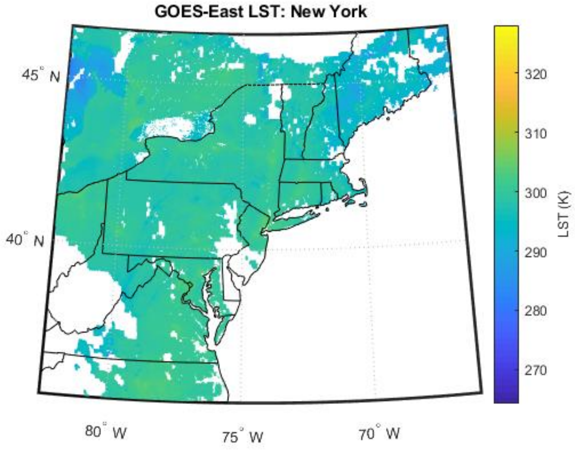
We can also use the Mapping Toolbox to add a shapefile delineating the borders between states. We can read a shapefile using 'shaperead' and then show it with 'geoshow'. Here we have downloaded a shapefile from https://www.census.gov/geo/maps-data/data/cbf/cbf_state.html

```
land = shaperead('shapefiles\cb_2017_us_state_20m.shp', 'UseGeoCoords', true, 'BoundingBox', [-
109.06, 25.32; -92.6, 37.93]);

figure(5)
worldmap([25.32, 37.93], [-109.06, -92.6])
pcolorm(Lat, Lon, GOES_LST)
h = colorbar; set(get(h, 'label'), 'string', 'LST (K)')
geoshow(land, 'DisplayType', 'polygon', 'FaceColor', 'none');
title("GOES-East LST: Texas")

figure(6)
worldmap([35.35, 46.53], [-82.51, -66.06])
land = shaperead('shapefiles\cb_2017_us_state_20m.shp', 'UseGeoCoords', true, 'BoundingBox', [-
82.51, 35.35; -66.06, 46.53]);
pcolorm(Lat, Lon, GOES_LST)
h = colorbar; set(get(h, 'label'), 'string', 'LST (K)')
geoshow(land, 'DisplayType', 'polygon', 'FaceColor', 'none');
title("GOES-East LST: New York")
```


GOES-East LST: Texas


GOES-East LST: New York

There are many places online where you can search GIS datasets. A good starting point is: https://freegisdata.rtwilson.com/

Published with MATLAB® R2018b