

# Single day plotting

In [3]:

```
1 import os
2 # Create a path to the pre-fire MODIS h4 data
3 modis_pre_path = os.path.join("E:\\",
4                                "MODIS_ET",
5                                "MOD16A2.A2018161.h25v07.006.2018179103116.hdf")
6 modis_pre_path
```

Out[3]:

```
'E:\\MODIS_ET\\MOD16A2.A2018161.h25v07.006.2018179103116.hdf'
```

In [4]:

```

1 from osgeo import gdal
2 import pyproj
3 import numpy as np
4
5 FILE_NAME = modis_pre_path
6 DATAFIELD_NAME = 'ET_500m'
7 GRID_NAME = 'MOD_Grid_MOD16A2'
8 gname = 'HDF4_EOS:EOS_GRID:"{0}":{1}:{2}'.format(FILE_NAME,
9                                                  GRID_NAME,
10                                                  DATAFIELD_NAME)
11 gdset = gdal.Open(gname)
12 data = gdset.ReadAsArray().astype(np.float64)
13
14
15 # Construct the grid.
16 x0, xinc, _, y0, _, yinc = gdset.GetGeoTransform()
17 nx, ny = (gdset.RasterXSize, gdset.RasterYSize)
18 x = np.linspace(x0, x0 + xinc*nx, nx)
19 y = np.linspace(y0, y0 + yinc*ny, ny)
20 xv, yv = np.meshgrid(x, y)
21
22 # In basemap, the sinusoidal projection is global, so we won't use it.
23 # Instead we'll convert the grid back to lat/lons.
24 sinu = pyproj.Proj("+proj=sinu +R=6371007.181 +nadgrids=@null +wktext")
25 wgs84 = pyproj.Proj("+init=EPSG:4326")
26 lon, lat= pyproj.transform(sinu, wgs84, xv, yv)
27
28 # Read the attributes.
29 meta = gdset.GetMetadata()
30 long_name = meta['long_name']
31 units = meta['units']
32 _FillValue = float(meta['_FillValue'])
33 scale_factor = float(meta['scale_factor'])
34 valid_range = [float(x) for x in meta['valid_range'].split(', ')]
35
36 del gdset
37
38 invalid = np.logical_or(data > valid_range[1],
39                          data < valid_range[0])
40 invalid = np.logical_or(invalid, data == _FillValue)
41 data[invalid] = np.nan
42 data = data * scale_factor
43 # data = np.ma.masked_array(data, np.isnan(data))

```

C:\Users\mohan\anaconda3\envs\osgeo\lib\site-packages\pyproj\crs\crs.py:141: FutureWarning: '+init=<authority>:<code>' syntax is deprecated. '<authority>:<code>' is the preferred initialization method. When making the change, be mindful of axis order changes: <https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-proj-6> (<https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-proj-6>)

in\_crs\_string = \_prepare\_from\_proj\_string(in\_crs\_string)  
C:\Users\mohan\AppData\Local\Temp\ipykernel\_22840\856647052.py:26: FutureWarning: This function is deprecated. See: <https://pyproj4.github.io/pyproj/stable/gotchas.html#upgrading-to-pyproj-2-from-pyproj-1> (<https://pyproj4.github.io/pyproj/stable/gotchas.html#upgrading-to-pyproj-2-from-pyproj-1>)

lon, lat= pyproj.transform(sinu, wgs84, xv, yv)

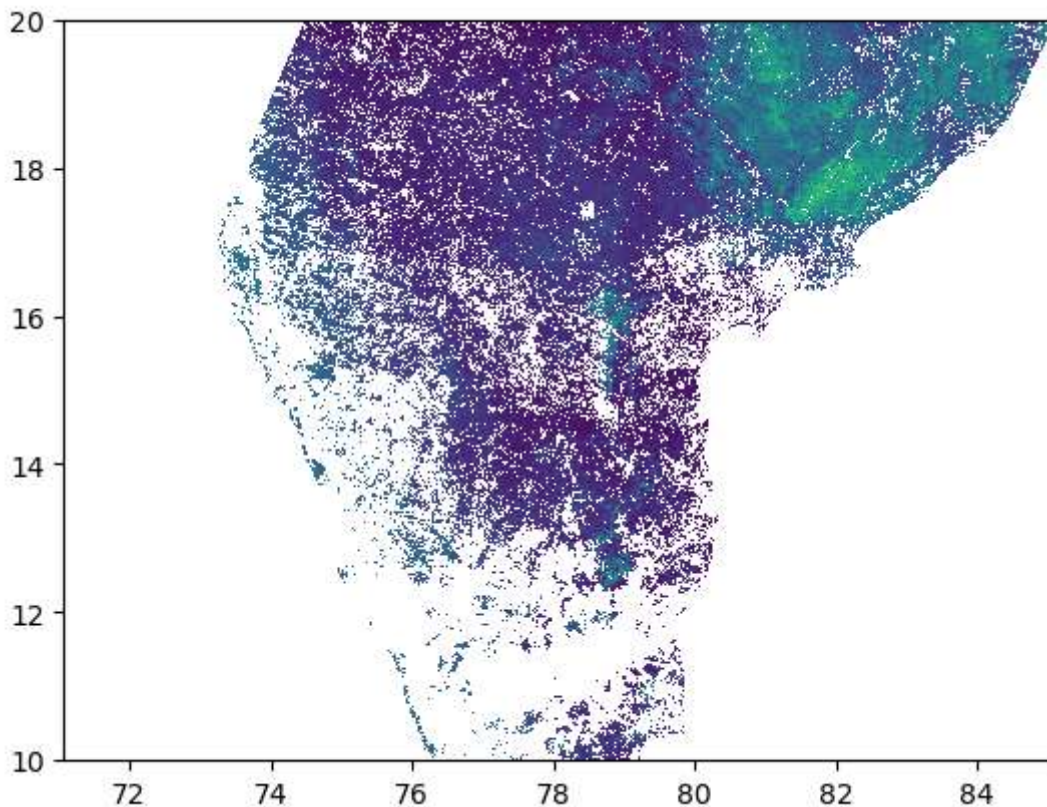
## Plotting using lon lat meshgrid made from above script (ideal case)

In [5]:

```
1 import matplotlib.pyplot as plt
2 plt.pcolormesh(lon,lat,data)
```

Out[5]:

<matplotlib.collections.QuadMesh at 0x1eba6c96b90>



## Plotting by creating lon lat arrays

In [6]:

```
1 lons= np.linspace(lon.min(),lon.max(),2400)
2 lats= np.linspace(lat.min(),lat.max(),2400)
3
```

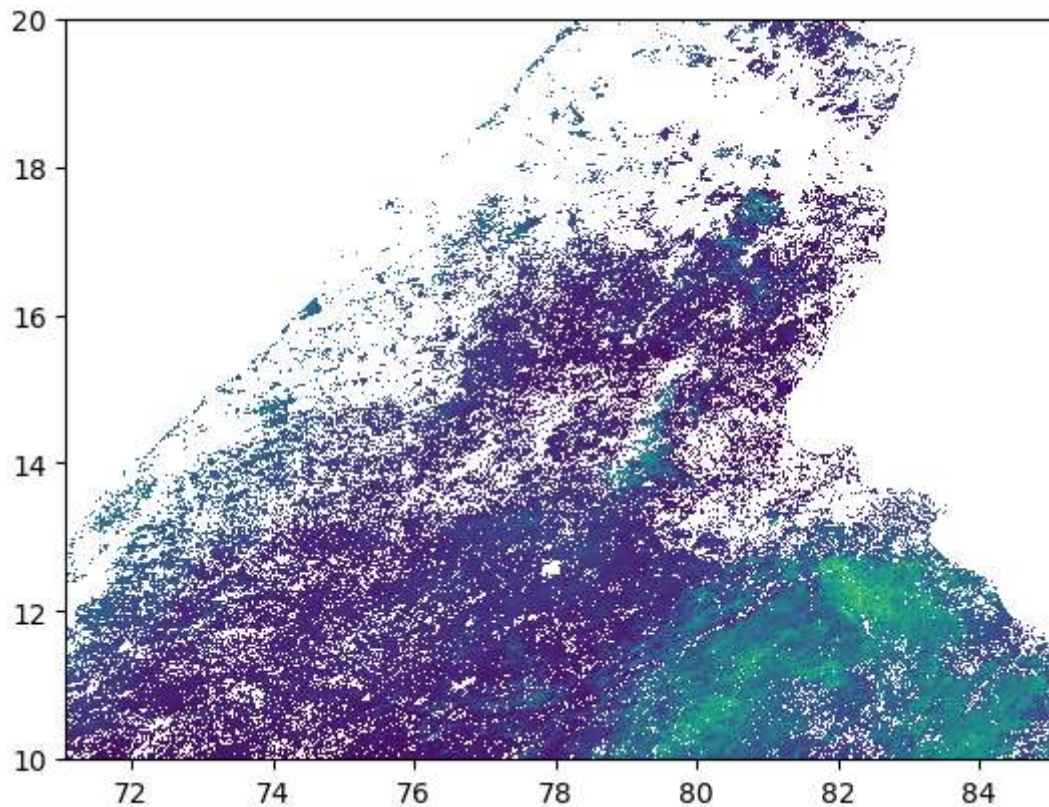
## The plot below appears to be inverted

In [8]:

```
1 plt.pcolormesh(lons,lats,data)
```

Out[8]:

<matplotlib.collections.QuadMesh at 0x1eba6e58730>



## Similar plot with flip

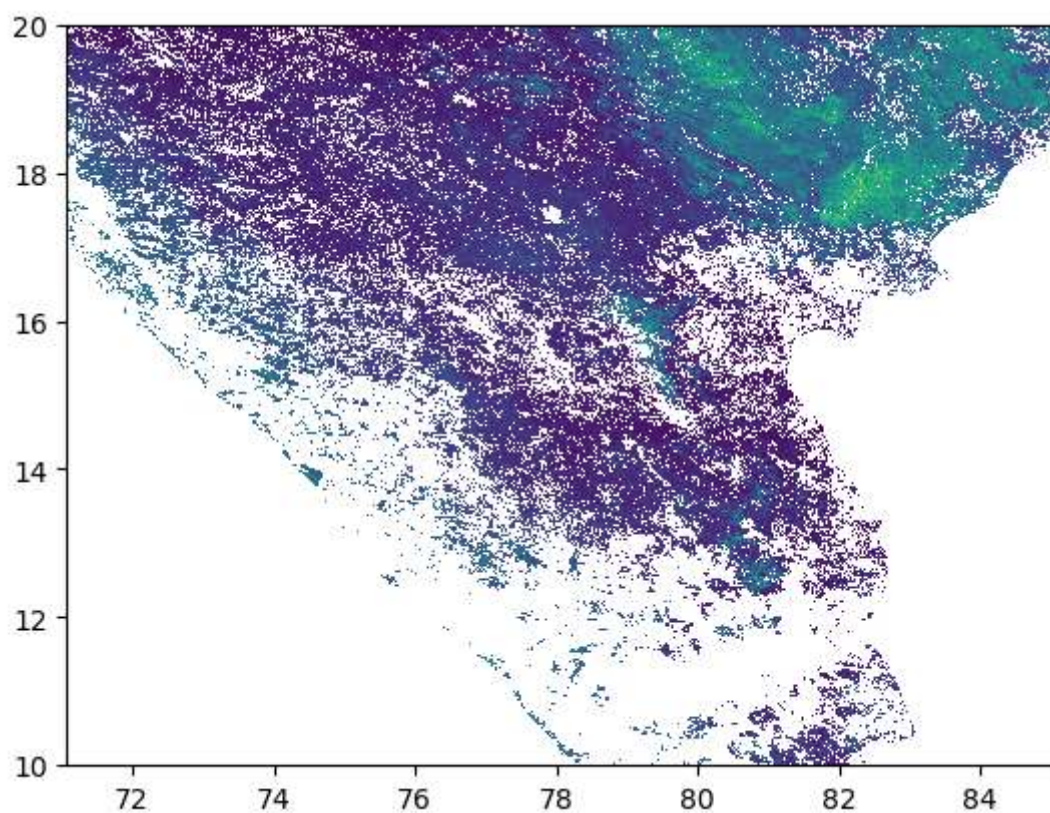
Here the plot is correct but leaning towards left of the viewer.

In [11]:

```
1 plt.pcolormesh(lons,lats,np.flip(data,axis=0))
```

Out[11]:

<matplotlib.collections.QuadMesh at 0x1eba7d27bb0>



In [ ]:

```
1
```