Untitled7

June 7, 2015

In [139]:
```
import numpy as np
import matplotlib.pyplot as plt


#Brute Force copied all array elements into a single txt file, replaces all "NaN" with 0
raw = np.loadtxt("C:/Users/IL/Documents/SciComp/HWK8/data2.txt")

#3d array where 3rd index is the month, "reshape" was doing something weird
data = np.zeros((4080/24, 180,  24))
for i in range(len(raw)):
    for j in range(180):
        data[i%170,j,int(i)/int(170)] = raw[i,j]

plt.imshow(data[:,:,23], origin = 'lower')
plt.show()
```

In [143]:
```
#Define Equatorial Pacific, Gulf of Mexico, North Indian Ocean, South Cali coast

#Average temperatures in the following regions for the 7 months
EPWarm = np.zeros(7) #11 degrees wide centered around equator
GMWarm = np.zeros(7) # (130,100)-(137,109), (bot left - top right)
NIWarm = np.zeros(7) # (24,80) - (47,95)
SCWarm = np.zeros(7) # (115,111)-(121,115)

temp1 =0
ctr1=0
temp2 =0
ctr2 =0
temp3 = 0
ctr3 = 0
temp4 = 0
ctr4 =0

#Warming Season

#Loop over each month
for i in range(2,9):
    #Scan each region, getting average
    for j in range(80,90,1):
        for k in range(180):
            if(data[j,k,i] != 0):
                temp1 += data[j,k,i]
                ctr1 += 1
    EPWarm[i-2] = temp1/float(ctr1)
```

```
        for j in range(100,109,1):
            for k in range(130,137,1):
                if(data[j,k,i] != 0):
                    temp2 += data[j,k,i]
                    ctr2 += 1
        GMWarm[i-2] = temp2/float(ctr2)

        for j in range(8,95,1):
            for k in range(24,47,1):
                if(data[j,k,i] != 0):
                    temp3 += data[j,k,i]
                    ctr3 += 1
        NIWarm[i-2] = temp3/float(ctr3)

        for j in range(111,115,1):
            for k in range(115,121,1):
                if(data[j,k,i] != 0):
                    temp4 += data[j,k,i]
                    ctr4 += 1
        SCWarm[i-2] = temp4/float(ctr4)

        temp1 =0
        ctr1=0
        temp2 =0
        ctr2 =0
        temp3 = 0
        ctr3 = 0
        temp4 = 0
        ctr4 =0
    print (EPWarm[6]-EPWarm[0])/len(EPWarm)
    print (NIWarm[6]-NIWarm[0])/len(NIWarm)
    print (GMWarm[6]-GMWarm[0])/len(GMWarm)
    print (SCWarm[6]-SCWarm[0])/len(SCWarm)

    #Plot of average temp by month. North Indian decreases cause it's in south hemisphere.
    plt.ylim(280,320)
    plt.plot(EPWarm,label = 'Equator')
    plt.plot(NIWarm,label = 'North Indian')
    plt.plot(GMWarm,label = 'Gulf of Mexico')
    plt.plot(SCWarm,label = 'South California')
    plt.legend(loc='upper left')
    plt.show()
```

```
0.115840763011
-0.423666921314
0.912536443149
0.729192546584
```

```
In [91]: print data[137,30,0]#Middle of Asia
         print data[30,137,0]#Coast of Chile
```

```
0.0
282.0
```

In [142]:
```
#Repeat for cooling season
EPCool = np.zeros(7) #11 degrees wide centered around equator
GMCool = np.zeros(7) # (130,100)-(137,109)
NICool = np.zeros(7) # (24,80) - (47,95)
SCCool = np.zeros(7) # (115,111)-(121,115)

temp1 =0
ctr1=0
temp2 =0
ctr2 =0
temp3 = 0
ctr3 = 0
temp4 = 0
ctr4 =0

#Warming Season

#Loop over each month
for i in range(9,16):
    #Scan each region, getting average
    for j in range(80,90,1):
        for k in range(180):
            if(data[j,k,i] != 0):
                temp1 += data[j,k,i]
                ctr1 += 1
    EPCool[i-9] = temp1/float(ctr1)

    for j in range(100,109,1):
        for k in range(130,137,1):
            if(data[j,k,i] != 0):
                temp2 += data[j,k,i]
                ctr2 += 1
    GMCool[i-9] = temp2/float(ctr2)

    for j in range(8,95,1):
        for k in range(24,47,1):
            if(data[j,k,i] != 0):
                temp3 += data[j,k,i]
                ctr3 += 1
    NICool[i-9] = temp3/float(ctr3)

    for j in range(111,115,1):
        for k in range(115,121,1):
            if(data[j,k,i] != 0):
                temp4 += data[j,k,i]
                ctr4 += 1
    SCCool[i-9] = temp4/float(ctr4)

    temp1 =0
    ctr1=0
    temp2 =0
    ctr2 =0
    temp3 = 0
    ctr3 = 0
```

```
        temp4 = 0
        ctr4 =0

    print (EPCool[6]-EPCool[0])/len(EPCool)
    print (NICool[6]-NICool[0])/len(NICool)
    print (GMCool[6]-GMCool[0])/len(GMCool)
    print (SCCool[6]-SCCool[0])/len(SCCool)

    plt.ylim(280,320)
    plt.plot(EPCool,label = 'Equator')
    plt.plot(NICool,label = 'North Indian')
    plt.plot(GMCool,label = 'Gulf of Mexico')
    plt.plot(SCCool,label = 'South California')
    plt.legend(loc='upper left')
    plt.show()
```

```
0.0575471698114
0.407379679144
-0.792419825073
-0.568322981366
```

Overall, the equatorial plane doesn't change much, the North Indian changes opposite to the Gulf of Mexico and California since it is in the south hemisphere.

Finding maximum gradient for the first July. Note: 1000km is roughly 15 degrees

```
In [130]: from math import radians, sin, cos, asin, sqrt


          import os

          from pandas import *
          from pandas.io.parsers import read_fwf

          def dist(lat1,lon1,lat2,lon2):
              """
              Calculate the great circle distance between two points
              on the earth (specified in decimal degrees)
              """
              # convert decimal degrees to radians
              lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])

              # haversine formula to calculate the distance between two points on surface on the earth
              dlon = lon2 - lon1
              dlat = lat2 - lat1
              a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
              c = 2 * asin(sqrt(a))
              r = 3956 # Radius of earth in kilometers. Use 3956 for miles, 6371 for km
              return c * r
          print dist(1,0,16,0)
```

```
1035.67837813
```

Maximum east-west Gradient

```
In [141]: #y ranges from 85 to 125 for north hemisphere
          temp1 = 0
```

```
latMax = 0
longMax = 0
for j in range(180-15):
    for i in range(85,125,1):
        if(data[i,j,6] != 0 and data[i,j+15,6]!= 0): #Check the line doesn't end on land
            if(abs(data[i,j,6] - data[i,j+15,6])> abs(temp1)): #New Max?
                temp1 = data[i,j,6] - data[i,j+15,6]
                latMax = i
                longMax = j #gives left point of line
print temp1
print(latMax, longMax)
```

```
9.2
(119, 143)
```

East west max gradient is off North American East Coast

```
In [140]: #Max north south grad
          #y ranges from 85 to 125 for north hemisphere
          temp1 = 0
          latMax = 0
          longMax = 0
          for j in range(180):
              for i in range(85,125-15,1):
                  if(data[i,j,6] != 0 and data[i+15,j,6]!= 0): #Check the line doesn't end on land
                      if(abs(data[i,j,6] - data[i+15,j,6])> abs(temp1)): #New Max?
                          temp1 = data[i,j,6] - data[i+15,j,6]
                          latMax = i
                          longMax = j #gives southern point of 1000km line
          print temp1
          print(latMax, longMax)
```

```
20.8
(105, 73)
```

Maximum gradient for north south line is south of Japan

It is unsurprising that the gradients are not equal since just looking at the original Panoply image, you could see most of the ocean temperatures in the horizontal direction were uniform(i.e. most temperature contours were horizontal), therefore the north-south gradient would be larger since the Sun is not heating areas 1000km (vertically) away from each other equally.

```
In [ ]:
```