ENV 480: Climate Laboratory, Spring 2014

Assignment 2: Exploring the CESM model output
Due: Tuesday February 18 2014

Question 1:
Comparing the CESM control run to NCEP reanalysis data

The previous exercise contained lots of information about the CESM control run, including code to open and read the model output files, and make some plots of model variables. **The following assumes you have completed that exercise and have already opened all the necessary files.** Note that you need to be connected to the internet to read the data files.

To validate our control run against observations, we are going to use the NCEP Reanalysis data. *Reanalysis data is really a blend of observations and output from numerical weather prediction models. It represents our "best guess" at conditions over the whole globe, including regions where observations are very sparse*.

The necessary data are all served up over the internet. We will look at monthly climatologies averaged over the 30 year period 1981 - 2010.
The data catalog is here, please feel free to browse:
http://www.esrl.noaa.gov/psd/thredds/dodsC/Datasets/ncep.reanalysis.derived/catalog.html

1a) Surface air temperature

Surface air temperature from the CESM control run is stored in the variable called `'TREFHT'` (for 'reference height temperature'). Take a look at it:

```
print atm_ctrl.variables['TREFHT']
```

Note that the dimensions are (12, 96, 144), meaning 12 months, 96 latitude points, and 144 longitude points.

i) Make a plot of surface air temperature for the month of January. This code is what you need:

```
fig = figure()
ax = fig.add_subplot(111)
cax = ax.pcolormesh( lon, lat, atm_ctrl.variables['TREFHT'][1,:,:] )
cbar = colorbar(cax)
ax.set_title('Surface air temperature, January (CESM control run)')
ax.axis([0, 360, -90, 90])
ax.set_xlabel('Longitude'); ax.set_ylabel('Latitude');
ax.contour( lon, lat, topo.variables['LANDFRAC'], [0.5,0.5], colors='k');
```

where the index `[1,:,:]` means "give me all lat/lon values for month #1".

ii) Now make the same plot from the NCEP reanalysis data.

Surface air temperature is contained in a file called "air.2m.mon.1981-2010.ltm.nc", which is found in the directory "surface_gauss". Here's a link directly to the catalog page for this data file:
http://www.esrl.noaa.gov/psd/thredds/dodsC/Datasets/ncep.reanalysis.derived/surface_gauss/catalog.html?dataset=Datasets/ncep.reanalysis.derived/surface_gauss/air.2m.mon.1981-2010.ltm.nc
Now click on the OPeNDAP link. A page opens up with lots of information about the contents of the file. The "Data URL" is what we need to read the data into our Python session. This code opens the file and displays a list of the variables it contains:

```
ncep_air2m =
nc.Dataset("http://www.esrl.noaa.gov/psd/thredds/dodsC/Datasets/ncep.reanalysis.derived/s
urface_gauss/air.2m.mon.1981-2010.ltm.nc")

for v in ncep_air2m.variables: print v
```

The temperature data is called `'air'`. Take a look at the details:

```
print ncep_air2m.variables['air']
```

Notice that the dimensions are (12, 94, 192) -- meaning 12 months, 94 latitude points, 192 longitude points. Not the same grid as our model output!

Use this code to make the plot of surface air temperature for January:

```
lat_ncep = ncep_air2m.variables['lat'][:]
lon_ncep = ncep_air2m.variables['lon'][:]

fig = figure()
ax = fig.add_subplot(111)
cax = ax.pcolormesh( lon_ncep, lat_ncep, ncep_air2m.variables['air'][1,:,:] )
cbar = colorbar(cax)
ax.set_title('Surface air temperature, January (NCEP reanalysis)')
ax.axis([0, 360, -90, 90])
ax.set_xlabel('Longitude'); ax.set_ylabel('Latitude');
ax.contour( lon, lat, topo.variables['LANDFRAC'], [0.5,0.5], colors='k');
```

iii) Make a similar plot for the **annual mean** surface air temperature from NCEP reanalysis. Include the code you used to make the plot.

Hint: the function `mean( )` *takes averages over arrays. You want to take the average over the first dimension of the temperature array, which is time (i.e. calendar months). If you have an array called somearray with multiple dimensions, this code*
`mean( somearray, axis=0 )`
will take the average over the first dimension only.

iv) Make a plot of the **difference** between the **annual mean surface air temperature** in CESM and NCEP reanalysis.

You will need to take the annual mean of the CESM data just like you did for the NCEP reanalysis above.

You will also need to deal with the fact that the two datasets are on different grids! *What happens if you just try to subtract one annual mean field from the other?*

We will interpolate the CESM data onto the NCEP grid. Here is a handy function to do the interpolation. Enter it exactly as written:

```
def regrid( lon1, lat1, data, lon2, lat2 ):
        from scipy.interpolate import griddata
        nlat = lat2.size; nlon = lon2.size
        X1,Y1 = meshgrid( lon1, lat1 )
        X2, Y2 = meshgrid( lon2, lat2 )
        points_in = transpose( array( [X1.flatten(),Y1.flatten()] ) )
        points_out = transpose( array( [X2.flatten(), Y2.flatten()] ) )
        data_interp = griddata(points_in,data.flatten(),points_out).reshape(nlat,nlon)
        return data_interp
```

This function takes a data array, along with its latitude and longitude coordinates, and interpolates to new grid specified by `lon2`, `lat2`.

Try the following:

```
newarray = regrid( lon, lat, atm_control.variables['TREFHT'][1,:,:], lon_ncep, lat_ncep )
print size(newarray)
```

You should find that you have created an array with the same dimensions as the NCEP reanalysis grid (94 latitude points, 192 longitude points).

Using this regrid function, you should be able to interpolate the annual mean CESM surface air temperature to the NCEP reanalysis grid and plot the difference between the two datasets. Include the code you used to interpolate and generate the plot.

v) Comment on the temperature differences you found.

1b) Precipitation

Now you follow a similar procedure to plot a map of the difference in annual precipitation between CESM and the NCEP reanalysis.

i) Make a plot of total annual precipitation in CESM (in units of meters)

There are actually two different kinds of precipitation generated by the model, and they are stored in two different variables ‘PRECC’ and ‘PRECL’. Let’s take a look:

```
print atm_control.variables['PRECC'].long_name, atm_control.variables['PRECC'].units
print atm_control.variables['PRECL'].long_name, atm_control.variables['PRECL'].units
```

These are given as monthly mean precipitation rates in meters per second. So to get the total annual precipitation, you need to three things:

- take the annual average of the monthly values
- multiply by the number of seconds in a year to get values in meters
- add the two kinds of precipitation together.

Bonus point: This procedure for calculating annual total is slightly inaccurate. Why?

Include the code you used to manipulate the data and make the plot.

ii) Make a plot of total annual precipitation in NCEP reanalysis (in same units).

This code will open the necessary data file:

```
ncep_prate = nc.Dataset(
"http://www.esrl.noaa.gov/psd/thredds/dodsC/Datasets/ncep.reanalysis.derived/surface_gaus
s/prate.sfc.mon.1981-2010.ltm.nc")
```

The reanalysis has a single precipitation variable called `prate`, and it's in a different unit:

```
print ncep_prate.variables['prate'].long_name, ncep_prate.variables['prate'].units
```

You will see that the units are kg / m^2 / s, i.e. it is given as a mass flux of water per unit area. You can convert this to m / s by dividing by the density of water. Then make a plot of the annual total as above.

iii) Make a plot of the difference in annual precipitation between CESM and NCEP reanalysis.

You will need to use the `regrid()` function to interpolate the data as you did with the temperature data in part (a). Again, include the code you used.

iv) Comment on the precipitation differences you found.

Question 2:
Greenhouse warming in CESM

You will now look at the results of a greenhouse warming experiment with CESM. The results of the control run were used as initial conditions for a new model run in which atmospheric CO_2 was instantaneously doubled from 367 ppm to 734 ppm. This leads to a decrease in outgoing longwave radiation (through the greenhouse effect, which we will be looking at more carefully later in the course). The model is therefore in dis-equilibrium and needs to warm up.

2a): Adjustment to new equilibrium temperature

First you will look at the time evolution of the model after CO_2 doubling. I have simplified things by taking global averages of all the model output. This code will open a file containing time series of global mean atmospheric model variables:

```
atm_2xCO2_global = nc.Dataset(
“http://ramadda.atmos.albany.edu:8080/repository/opendap/latest/Top/Users/Brian
+Rose/CESM+runs/som_2xCO2/som_2xCO2.cam.h0.global.nc/entry.das " )
```

Take a look at the contents of this dataset, and convince yourself that it has all the same variables as the control run output, but different dimensions. For example, compare the output of

```
print atm_control.variables['TS']
```

and

```
print atm_2xCO2_global.variables['TS']
```

The global mean dataset has monthly average values over a period of 30 years, for a total of 360 data points per variable.

i) Make a plot of surface temperature versus time for the 2xCO_2 run. *You can plot either `'TS'` (the surface temperature) or `'TREFHT'` (the air temperature just above the surface). Why not try both?*

ii) Discuss the shape of the resulting plot. How is it similar to / different from the temperature adjustment curves we plotted for the simple energy balance model in Assignment 1?

iii) You may find that the global mean temperature in CESM varies with a well-defined annual cycle. Verify that this cycle is also found in the observations.

For this, you will need to compute the global average of the NCEP reanalysis surface air temperature data for each month. When taking spatial averages, we need to be careful about the fact that the area of each grid cell on a latitude-longitude grid is not constant (cells get smaller near the poles). Mathematically, we need weight the average by the cosine of the latitude (which approaches zero at the poles). This function does the trick:

```
def globalmean( lat, data ):
       return sum( mean( data, axis=1) * cos(deg2rad(lat)) ) / sum( cos(deg2rad(lat) ) )
```

This uses functions called `sum()` and `mean()` which perform sums and averages! What do you think `deg2rad()` does? Experiment with it to convince yourself.

So for example, to compute the observed global average surface air temperature for the month of January, try this:

```
print globalmean( lat_ncep, ncep_air2m.variables['air'][1,:,:] )
```

Show the seasonal cycle (12 months) of observed surface air temperature. You may present either a table of values or a graph. Please include the code you used to generate your answer (copy and paste).

iv) Speculate on why the global mean surface temperature varies throughout the year in this way.

2b): Compute the equilibrium climate sensitivity (ECS) for CESM.

Recall that we defined ECS as the global mean surface warming (a number in degrees K) in response to doubled CO_2 after adjustment to the new equilibrium.

You will need to take annual averages and global averages of the surface temperature before and after the CO2 doubling. Use the control run climatology as the "before" temperature. Here is the URL you will need to open the equivalent climatology file from the 2xCO2 run (which was created by taking time averages over the last 10 years of the simulation, i.e. after it has reached equilibrium):

```
"http://ramadda.atmos.albany.edu:8080/repository/opendap/latest/Top/Users/Brian+Rose/CESM
+runs/som_2xCO2/som_2xCO2.cam.h0.clim.nc/entry.das"
```

Also include the code you used to calculate ECS.