Lab 6: Python frontogenesis

Objective: To continue learning Python and how it could be used to diagnose frontogenesis.

Materials: Your laptop, Enthought Python, access to the Internet, a No. 2 black pencil, eraser, colored pencils,and a straight edge (ruler). It will not be necessary to log into the SoM machines if you have a laptop, however if you do not have a laptop you can also complete this lab on any of the SoM machines in room 5720.

Procedure:

1) Downloads
   (a) We will be using our GFS analysis Netcdf file from last week named `gfs_4_20120908_0000_000.nc`, so there are no new downloads.
   (b) No new or updated scripts to download from the class repository. You will be using your versions of `plot_gfs_fields_forlab.py` and `plot_gfs_cross_section.py`.
2) Plots
   Recall that the Frontogenesis equation is

$$F = \left[\frac{\partial\theta}{\partial x'}\left(\frac{\partial u}{\partial y'}\right)\right] + \left[\frac{\partial\theta}{\partial y'}\left(\frac{\partial v}{\partial y'}\right)\right] + \left[\frac{\partial\theta}{\partial p}\left(\frac{\partial\omega}{\partial y'}\right)\right] - \left[\frac{\partial}{\partial y'}\left(\frac{\partial\theta}{\partial t}\right)\right] \quad (1)$$

   where the $'$ symbol means the axes are using a **natural coordinate system.** In the above, the first term on the right-hand side is the *shearing term*, the second is the *confluence term*, the third is the *tilting term*, and fourth is the *diabatic term*.
   (a) Edit `plot_gfs_fields_forlab.py` to compute and plot potential temperature (in Kelvin) instead of sea level pressure and isotherms. To do this:
      (i) Add an option in the `user_options` section near the top that says `plot_theta = 'true'`
      (ii) Read in surface pressure (i.e. not reduced to sea level). This can be done in the `if ( level_option == -1 ) :` section where the variables are read in from the netcdf file by adding `pres0m = f.variables['PRES_P0_L1_GLL0'][::-1,:]`.
      (iii) Convert temperature to Kelvin, then convert to potential temperature. To do so, under the comment `Convert temperature to degrees Celsius` and under `if (level_option == -1 ) :`, edit your script so that the block says:

```
if (plot_theta == 'true') :
    temperature_plot = temperature_plot + 273.15 # Convert to Kelvin
    temperature_plot = wm.temp_to_theta(temperature_plot, pres0m)
else:
    temperature_plot = (temperature_plot * (9./5.)) + 32
```

      (iv) Edit your contour interval block so that it looks like the following:

```
if ( level_option == -1 ) :
   if (plot_theta == 'false'):
      cflevs_temp = np.arange(-40,101,4)
   else:
      cflevs_temp = np.arange(200,401,4)
```

      (v) Edit your plot so that it does not plot sea level pressure when the `plot_theta` option is `true`:

```
if ( plot_contours == 'true' ):
   if (plot_theta == 'false'):
      CS = m.contour(x, y, plotvar, cflevs, colors='k', linewidths=1.5)
      plt.clabel(CS, inline=1, fontsize=10, fmt='%i')
```

(vi) When plotting at 500 hPa, 850 hPa, or the surface, set it so that you plot potential temperature contours using a solid black contour only when `plot_theta` option is `true`:

```
if ( plot_theta == 'true'):
   CS2 = m.contour(x, y, temperature_plot, cflevs_temp, colors='k', \
         linestyles='solid',linewidths=1.5)
else:
   CS2 = m.contour(x, y, temperature_plot, cflevs_temp, colors='r', \
         linestyles='dashed',linewidths=1.5)
```

(vii) Run your script. If it plots a surface map with wind barbs and potential temperature contours (in Kelvin), then you are ready to move to the next step.

(b) Add the following in the **user options** section of `plot_gfs_fields_forlab.py`:

```
latpts  = [34.00, 38.80, 36.0]
lonpts  = [-98.50, -89.0, -107.5]
```

Then add the following:

```
xpts,ypts = m(lonpts,latpts)
m.plot(xpts,ypts,'bo')
```

near the end of your plot, just before you add the title and save the image. Execute the script again. You should get 3 points plotted in addition to your wind barbs and potential temperature contours. These 3 points should be located in northwestern New Mexico, north central Texas, and southern Illinois. Print your image.

(c) Using a ruler, draw out your x' and y' axes so that y' is perpendicular to the front and x' is along the front. Measure both axes to be exactly 1 inch in total length (0.5 inches on each side of the point). Do this for all 3 points.

(d) Use the web site `http://www.mapdevelopers.com/distance_finder.php` to estimate the physical distance along your axes in meters.

(e) Estimate the *confluence* and *shearing* terms of the frontogenesis equation (Equation (1)) using finite differencing at each of the three points. For example, to compute $\frac{\partial a}{\partial y'}$, calculate $\frac{a_2 - a_1}{\Delta y'}$ where a_2 is the value of a at the top of your y' axis, a_1 is the value of a at the bottom of your y' axis, and $\Delta y'$ is the physical distance of your y' axis. Use scratch paper for your calculations, then fill in the table below to keep track of your results. You do not need to turn in your scratch sheet of paper.

(f) Based on what you have found, shade surface frontogenesis on your surface plot that is due to either confluence or shearing. Finalize using a colored pencil with the color of your choice.

(g) Add vertical motion to `plot_gfs_cross_section.py`:

(i) Open your version of `plot_gfs_cross_section.py`

(ii) Make sure the following user options are set:

```
date_string = '2012090800'
varname2fill = 'VVEL_P0_L100_GLL0'
```

Calculation	Point 1	Point 2	Point 3
$\Delta x'$ (meters)			
$\Delta y'$ (meters)			
$\frac{\partial \theta}{\partial x'}$			
$\frac{\partial u}{\partial y'}$			
$\frac{\partial \theta}{\partial y'}$			
$\frac{\partial v}{\partial y'}$			

Table 1: Use this table to keep track of your frontogenesis estimates.

```
varname2cntr ='TMP_P0_L100_GLL0'
plot_theta = 'true'
plot_temperature = 'true'
plot_anomaly = 'false'
plot_winds = 'false'
plot_contours = 'true'
lat_range = [25,50]
lon_range = [261.5,261.5]
pres_range = [200,1000]
figname = "gfs_cross_section_analysis_ew_lab6"
```

(iii) If it's not already, fpath should be set to the directory of your GFS netcdf file from last week's lab.

(iv) Add the following *if statement* in its own block in section where the base contour, contour interval, and number of contours are set:

```
if varname2fill == 'VVEL_P0_L100_GLL0':
   base_cntr = 0 # Base contour
   cint = 0.2 # Contour interval
   nconts = 10 # number of contours
```

(v) Add the following *if statement* in its own block just above `label_fontsize = 14`:

```
if varname2fill == 'VVEL_P0_L100_GLL0':
   djet = djet2
```

(vi) Run your `plot_gfs_cross_section.py`. If everything has been done correctly so far, you will get a lot of messages printed to the screen, including one that says something like "TypeError: Length of x must be number of columns in z, and length of y must be number of rows.". This error occurs because although ω is on isobaric levels, it does not have the same number of isobaric levels as variables we have previously plotted, such as temperature and wind. We need to add a block to the section where variables are read in from the netcdf file near the top of your script as follows:

```
if (varname2fill == 'VVEL_P0_L100_GLL0' ):
   omega = f.variables['VVEL_P0_L100_GLL0'][:,::-1,:]
   omega_plot = np.zeros_like(temperature).astype('f')
   omega_plot[0:4,:,:] = float('NaN')
   omega_plot[5:,:,:] = omega[:,:,:]
   if (varname2fill == 'VVEL_P0_L100_GLL0' ):
      var2fill = omega_plot
```

(vii) Now re-run your script. If successful, you will get a cross section from south to north along a longitude line that cuts through the point in north central Texas on your surface map from earlier. For this plot, assume that your x-axis (which is latitude here) represents your y' axis, where y' is increasing northward. Here, we will not attempt to quantify values for frontogenesis from the titling term, but will instead do so qualitatively. Briefly describe where the strongest *ascent* is located in relation the the surface front, and whether you think the tilting term of the Frontogenesis Equation is contributing to frontogenesis based on this plot. Note that your surface front may extend vertically upward, meaning that it is possible to have frontogenesis on multiple levels. You do not need to print this plot.

(viii) For the point located in southern Illinois, let the y' axis be oriented from west to east through that point. Adjust your user options and re-run. Note that you may need to round your latitudes in `lat_range` to the nearest integer to correspond to the nearest GFS data point. Add anything you find relevant to your discussion above.

(ix) Repeat for the point in New Mexico, letting the y' axis be oriented from west to east through that point. Add anything you find relevant to your discussion above.

,

Hand in all plots for this lab with answers to any questions above. This lab is due at the beginning of class on Wednesday September 26.