AOS 452 Lab 5 Handout
GEMPAK Part II – Gridded Data Programs

Introduction

This lab will concentrate on working with gridded data sets, so we will be using the GD- (**g**ridded **d**ata) programs. The GEMPAK programs available for working with gridded data sets (usually model data) are as follows:

gdinfo list information about the variables contained in a gridded data set
gdcntr create horizontal plots of scalars (display contours)
gdwind create horizontal plots of vectors (display barbs or arrows)
gdplot a combination of `gdwind` and `gdcntr`
gdcross create vertical cross sections using gridded data
gdprof create vertical profiles using gridded data

The gridded data sets available in real-time can be found in the **$MODELDATA** directory (or /weather/data/gemdata/hds). The files in this directory are named **yymmddhh_model.gem** where **yymmddhh** is the 2-digit year, month, day, and hour (in UTC) respectively, and **model** identifies which numerical model output is contained in the file. Possible choices for **model** include:

thin_avn Global Forecast System (GFS) on 2.5 degree lat/lon global grid
avn003 GFS on 1 degree lat/lon global grid
avn211 GFS on high-res CONUS grid (The "211 Grid", 80 km grid spacing)
ecmwf ECMWF European global model
eta NAM model on 211 Grid (formerly known as the Eta model)
eta104 NAM model on 104 Grid (covers all of N. America, 90 km grid spacing)
nam212 NAM model on 212, 215, 218 Grid (covers all of N. America with 215 or 218 different grid spacings)
ruc Rapid Update Cycle (short-term forecasts on 211 Grid)
ruc40 Rapid Update Cycle (short-term forecasts with 40 km grid spacing)

Note: Full specifications for the 104 Grid, or any other official grid, can be found at
http://www.nco.ncep.noaa.gov/pmb/docs/on388/tableb.html, or
http://www.emc.ncep.noaa.gov/mmb/namgrids/

`GDINFO` – Gridded Data Information

To list information contained in a GEMPAK gridded data file, one can use the GEMPAK program gdinfo. To begin using this program,

type gdinfo at the Unix prompt.

Let's look at yesterday's NAM model output. Make sure the parameters are set as follows:

GDFILE = $MODELDATA/17092012_eta.gem (or: /weather/data/gemdata/hds/17092012_eta.gem)
LSTALL = YES
OUTPUT = T
GDATTIM = all
GLEVEL = all
GVCORD = all
GFUNC = all

This will open the 1200 UTC 20 September 2017 NAM model GEMPAK file and output its contents to the terminal window. Once you change the above parameters, type *r* and press enter to run the program. First, the file name and grid information are shown. Then, a listing of all of the available variables, vertical levels at which each variable can be found, and which forecast times are available is shown. To page through the listing, continue to hit enter until the listing is complete.

It is important to know what variables are available in a GEMPAK file. Depending on the file, it may not contain a variable that you want to plot. However, one of the great features of GEMPAK is that it can plot some variables even if they are not explicitly provided in the GEMPAK file using the variables that are included in the file. For example, the ageostrophic wind is not explicitly provided in many GEMPAK files. However, GEMPAK can derive this quantity by using geopotential height and wind fields that are provided in the GEMPAK file.

GDCNTR – Gridded Data Contours (Plots contour maps of scalar variables)

The program gdcntr plots scalar variables (if they are available in or derivable from a GEMPAK file) such as temperature, height, sea level pressure, etc. In addition, you can plot combinations of the available variables such as vorticity, temperature advection, thickness, and divergence. Many of the parameters in gdcntr are the same as those described in Lab 4. Here is a listing of some important ones that differ:

gdattim:	Specifies date/time to plot
glevel:	Vertical level to plot
gvcord:	Vertical coordinate in which the *glevel* is defined
gfunc:	Scalar function to plot/contour
cint:	Contour interval to use
line:	Defines line type, color, thickness, etc.
gdfile:	Gridded data GEMPAK file to use
ctype:	Defines whether to contour or use color filling
scale:	Scaling factor to plot variables of order one

I strongly suggest reading up on the *gfunc* parameter (***phelp gfunc)*** to get an idea of the variety of variables that can be plotted using GEMPAK.

Let's begin by plotting the 12 hour forecast of 500 mb geopotential height from the 1200 UTC Eta model run yesterday morning ($MODELDATA/17092012_eta.gem). Change the following parameters as outlined in addition to any other changes that may be necessary:

```
Gdfile = $MODELDATA/17092012_eta.gem (or: /weather/data/gemdata/hds/17092012_eta.gem)
gdattim   = f12                  ← This is where to change the forecast hour.
glevel    = 500
gvcord    = pres
gfunc     = hght
cint      = 60
line      = 3/1/2/1              ← This uses line color 3/type 1/thickness 2/labeled
map       = 8
device    = xw
garea     = 20;-125;55;-55       ← Make a note of these garea and proj settings, as
proj      = lcc/20;-97.5;55        they are good for CONUS plots in general.
clear     = yes
panel     = 0
scale     = 0
ctype     = c
title     = 1/-2/~ Your name here  ← The ~ adds the date/time/forecast hour
text      = 0.75                     and @ adds the glevel to the title.
skip      = 0
latlon    = 0
```

Once you have changed these parameters, type `r` and press `<Enter>`. After you get that plot to work correctly, try changing ***gdattim*** to look at the 48-hour forecast 500-mb geopotential height.

The ***scale*** parameter is used to multiply variables by a certain factor of 10. As an example, let's say we wanted to plot vorticity, where values are typically on the order of 10^{-5} s^{-1}. We would then want to set scale = 5, so that the vorticity gets multiplied by 10^5 before being plotted. Likewise, let's say we wanted to plot geopotential height with units of decameters (dam). We would then need to set scale = -1.

The ***line*** and ***cint*** parameters can be manipulated such that a zero line is omitted and negative contours are dashed. To show this, let's plot the 24-hour forecast 850 mb potential temperature advection by the total wind. First, change the following parameters:

```
gdattim   = f24
glevel    = 850
gvcord    = pres
gfunc     = adv(thta,wnd)
line      = 3/1/2/1
cint      = 1/1                  ← This says contour interval of 1 with a min of 1.
```

```
scale      = 4              ← This multiplies values by 10^4
```

Once you get this plot to work correctly, (by entering `run`), immediately change the following parameters:

```
cint       = 1//-1          ← This says contour interval of 1 with a max of -1.
line       = 3/2/2/1        ← Now using line type 2.
clear      = no
```

Run the program again. Now you should have a plot without a zero contour. In addition, the positive values should be contoured with solid lines, and negative values with dashed lines. Exit the program, and type **gpend.**

GDWIND – Gridded Data Vectors

The GEMPAK program **gdwind** is similar to **gdcntr**, except it plots vectors (wind, for example) instead of scalars. The parameter list looks similar to that of **gdcntr**, but there are some differences. Instead of *gfunc*, we now have *gvect* (for vector). To see how this program works, set the following parameters to plot the 500-mb wind (using the same *gdfile* as before):

gdfile = $MODELDATA/17092012_eta.gem (or: /weather/data/gemdata/hds/17092012_eta.gem)
gdattim = f12
glevel = 500
gvcord = pres
gvect = wnd
wind = bk1/0.75 ← This plots (b)arbs in (k)nots in color 1 and size .75
device = xw
clear = yes
scale = 0

Once you get the parameters set correctly, you should see a plot of wind barbs at 500 mb. If it looks too cluttered, try using the skip parameter to get rid of some of the wind barbs (e.g., **skip** = /2). Now plot the ageostrophic wind instead of the real wind for the same time and level.

GDPLOT – Gridded Data Plotting

The GEMPAK program **gdplot** allows for the display of both scalar and vector fields. In essence, gdplot is a combination of both gdcntr and gdwind. Start gdplot and get a list of the parameters. Notice that this long parameter list is a combination of parameter lists from both gdwind and gdcntr. To become familiar with gdplot, create a plot of 500-mb absolute vorticity and wind (barbs in knots), using the 12-hour forecast from the same Eta GEMPAK file we have used throughout the lab (1200 UTC 20 September 2017). Exit gdplot and type gpend.

Note: A more powerful version of **gdplot**, called **gdplot2**, is also available. You may want to experiment with that program as well. Use `phelp gdplot2` to see what the differences are.

GDCROSS – Gridded Data Cross Sections

The GEMPAK program **gdcross** can be used to create vertical cross sections from gridded data. Here are a few new parameters in **gdcross**:

cxstns:	Defines the endpoints of the line through which the cross section is taken
ptype:	Type of y-axis used in drawing the cross section
yaxis:	Defines the format and labeling frequency for the y-axis
border:	Background properties

For **gdcross**, the most important parameter to grasp is *cxstns*. There are several ways to define the cross section line in **gdcross** (note that these ways WILL NOT work in **sncross**):

1. Specify two three-letter identifiers serving as the endpoints of the line (e.g., `lax>bwi`).
2. Use the `cursor` command to graphically draw the cross section line from a plot in an x-window when you are using gdplot or gdcntr or gdwind.
3. Specify the latitude/longitude coordinates of the endpoints.

Enter **gdcross**, and change the following parameters:

```
cxstns    = 45;-140>45;-90
gvcord    = pres
gfunc     = thta
gvect     = wnd
ptype     = log
line      = 3/1/2/1
wind      = bk1/1.0
yaxis     = 1000/200/100
gdattim   = f12
cint      = 4
```

Run **gdcross** with these settings, and you should see wind barbs representing the horizontal wind direction and speed at a given pressure level and contours of the potential temperature field. Pressure is labeled along the left axis, while theta is labeled along the right axis.

***CURSOR** command*

The `cursor` command in any of the GEMPAK programs is a useful tool to re-define the *garea* (in **gdplot** or **gdcntr**) to zoom into a feature of interest or define a cross section line for **gdcross**. First, exit **gdcross** and re-enter **gdplot** (or **gdcntr**). Let's go through some examples of how to use the `cursor` command.

CURSOR GAREA

1) Create a plot of 850-mb temperature while having *device* = xw.
2) Once your plot is in the x-window, type the command `cursor garea`.
3) Move the cursor over the graphics window until a plus sign appears. Once you see the plus sign, draw a box around a portion of the domain shown by holding down the left mouse

button as you extend the box from one corner to the opposite corner. Release the mouse button once you are satisfied. An outline of the new graphics area will be shown as you click and drag the mouse. If you are unhappy with the box, just re-type `cursor garea` at the prompt and redraw your box.

4) Once you have the box as you want it, execute the plot in **gdplot** (or **gdcntr**) by typing `r` or `run`, and *garea* will automatically have been redefined for you. The new graphics area shown will be the one that you defined with your box.

CURSOR CXSTNS

In much the same way, you can use the command `cursor cxstns` to define a cross section line. The `cursor cxstns` command is executed in any program that *plots data on a horizontal plane,* such as **gdcntr**, **gdwind**, or **gdplot**. If you want to analyze a vertical cross-section over a particularly interesting region while working in one of these programs, follow these steps:

1) Plot, for example, 850-mb temperature in **gdplot**.
2) Type `cursor cxstns` at the prompt. The red arrow cursor will change to a white plus sign.
3) Move the white plus sign to the location that you want as the left endpoint of the cross section.
4) Press and hold down the left mouse button, then drag the cursor to the location that you want as the right endpoint of the cross section. As you drag the cursor, a line along which the cross section will be drawn appears on the screen.
5) Release the mouse button when you reach your desired second endpoint. The line along which the cross section will be drawn will remain on the screen. If you are not satisfied with the line you have drawn, repeat steps 2–5.
6) Once you are satisfied with the line, exit **gdplot** (or **gdcntr**) without typing `gpend`.
7) Enter the **gdcross** program, and type `l` to get a listing of your parameter settings. Notice that the cross section line (*cxstns*) has already been changed.

GDPROF – Gridded Data Vertical Profiles

Vertical profiles of gridded data can be created using the program **gdprof**. If you do not have observational data to view a vertical profile, you can use **gdprof** to get a vertical profile derived from model data. Most of the parameters in **gdprof** are the same as those in **snprof**. There are only two new parameters we need to cover:

gpoint: The *gpoint* parameter in **gdprof** works much the same way as the *area* parameter in **snprof**. The *gpoint* parameter can be used to define a location in the following ways:

1. By the grid point number. Example: `gpoint = @2;2` means at grid point (2,2)
2. By a latitude/longitude pair. Example: `gpoint = 50;-140` means at 50°N, 140°W
3. By station identifier. Example: `gpoint = msp` means at the point of the MSP (Minneapolis) station

4. Using the CURSOR command

We will be using method #4 in this lab to define the *gpoint* parameter.

output: Numerical data from the model vertical profile will be printed to the device given in the *output* parameter. Two possible values for *output* that you likely will use the most are the following:

`T`	Output goes to the terminal window
`F/name_of_file.dat`	Output goes to the file *name_of_file.dat*

CURSOR GPOINT

In a previous section, we learned about two `cursor` commands, `cursor garea` and `cursor cxstns`. We will now learn a third `cursor` command, `cursor gpoint`. As with the other two `cursor` commands, you will be using the `cursor gpoint` command in GEMPAK programs that plot data on a horizontal plane (e.g., **`gdcntr`**, **`gdplot`**).

1. Go into **`gdplot`** and create a plot of 850-mb temperatures over the United States.
2. Once the plot is in the x-window, type `cursor gpoint` at the prompt.
3. Move the cursor into your graphics window until a white plus sign appears.
4. Move the cursor to a desired location with a relatively strong temperature gradient and click the left mouse button. You have just chosen the point for the vertical profile in **`gdprof`** even though we are in **`gdplot`**. A small "X" may appear at the point you selected.
5. Exit **`gdplot`** *without* typing `gpend`.
6. Start **`gdprof`** and take a look at the parameters. The parameter *gpoint* has a latitude/longitude pair that was set through your use of the `cursor gpoint` command.

Now that we are in **`gdprof`** with *gpoint* set for some desired location, let us create a vertical profile derived from the model data. Set the following parameter values:

```
gvect     = wnd
gvcord    = pres
gfunc     = tmpc
gdfile = $MODELDATA/17092012_eta.gem (or: /weather/data/gemdata/hds/17092012_eta.gem)
line      = 2/1/4/1
marker    = 0
border    = 1
ptype     = skewt
scale     =
xaxis     = -40/40/10
yaxis     = 1000/100/100
wind      = bk1
refvec    =
winpos    = 1
filter    = 0.5
```

```
title     = 1
panel     = 0
clear     = yes
text      = 0.75
device    = xw
output    = T
thtaln    = 7/1/1/1
thteln    = 9/1/1/1
mixrln    = 12/2/1/1
```

Run the program. A skew-T diagram for the point you specified in **gdplot** should appear. Temperature data from various levels also should appear in the terminal window.

Unlike in **snprof** where multiple variables can be plotted at one time, you can only plot one scalar and/or vector variable at a time in **gdprof**. In order to add the dewpoint temperature to the current plot, for instance, we must change a few parameters and run **gdprof** again. Change the following parameters:

```
gfunc     = dwpc
line      = 4/1/4/1
clear     = no
```

Run the program again. You should now have a plot with both temperature and dew point temperature. (Setting `clear = no` prevented the temperature line from being erased.) Exit **gdprof** and use `gpend`.

You now have the tools to begin incorporating diagnostic maps into your own weather discussions. You can display these maps on your own webpage, which we will make in a couple weeks. Start thinking about what types of maps you'd want to include in your weather discussions so that you are not rushed once it gets here. In the next few GEMPAK labs we will learn some tools to make these maps more presentable. If you need some help, don't hesitate to ask!