

AOS 452 - Lab 6 Handout

GEMPAK III: Hints and Tricks

INTRODUCTION

By now you have had experience with a few of the key GEMPAK programs. This likely means that you have also had experience with the necessary process of tweaking parameters in order to make presentable plots within these programs. If you are anything like me, it seems like you spend more time picking the right colors and contour intervals for your figures than you take to enter the parameters to produce those figures. This lab contains even more information about creating presentable figures and ways to make your figures look super fancy. Also, this lab will give you a brief overview of a few more GEMPAK programs (YAY), in addition to a few useful variables that we have overlooked in past programs. You are not required to use these extra programs for class, but they are excellent tools to create presentation-quality figures.

GDPLLOT2 (essentially the same as GDPLLOT):

Draws contour lines through scalar grids and/or wind barbs, arrows, or streamlines through vector grids. Multiple sets of contours, vectors and/or streamlines can be generated for each frame.

Some of the unfamiliar variables in GDPLLOT2:

GDPFUN - Scalar grid or vector grid function (a combination of gfunc and gvect).

In this program you can plot multiple variables at one time, which saves time in GEMPAK. How do you do this? You can use exclamation points to separate variables. For example:

```
GDPFUN = tmpc ! wnd ! knts(mag(wnd))
```

will plot temperature, wind barbs, and isotachs on the same map, during the same run. This means that you will have to use the “!” when setting your cint and line, or fint and fline. This can be done by using the “!” in the same manner.

TYPE - GDPLLOT2 function processing type (can be defined for both scalars and vectors).

You are already familiar with this command but here is an advanced look at type. Did you know you could define multiple types for one variable? For example, if we assumed the gdpfun above, we could set

```
TYPE = c ! b ! c/f,
```

where the temperature is contour lines, the wind plotted with barbs, and the magnitude of the wind in knots is contoured and filled. These are the options for type:

| <u>SCALAR TYPES:</u> | <u>VECTOR TYPES:</u> |
|--|----------------------|
| C the original GEMPAK contouring algorithm | A wind arrows |
| F contour fill algorithm | B wind barbs |
| P plot grid point values | S streamlines |
| D plot scalar as a directional arrow | |

If you notice, GEMPAK plots the variables in the order that you state, so make sure you have your color fill variables plot first in your presentation figures.

LINE - Here is a helpful hint, you can plot negative values with a dashed line if you use `line = 3/-2`, where the minus sign in front of the 2 tells GEMPAK to use dashed lines for negative values only.

HILO - Allows you to plot the maximum and minimum values of a variable (useful for sea level pressure or jet streaks).

We can find the strongest winds in a domain and plot symbols at the point. For example, for our `gdpfun` we set above, we can set

```
HILO = !! 2;/J#;/120-180;
```

Where we are following this algorithm to set `hilo = (color of the high);(color of the low)/(symbol for the high);(symbol for the low)/(range to search for the high);(range to search for the low)`. The “#” after the “J” means that the numerical integer value will be plotted below the symbol.

HLSYM - This parameter works with `hilo` and allows you to set the text size, position and font of the symbol and numbers in this manner: `text size/position/font type(1-4)`. See `phelp hlsym` for more information.

CLRBAR - When you use a fill pattern, the use of a color bar is beneficial. This is the color bar syntax: `color / orientation / anchor / x;y / length;width /frequency | text_info`, you can read more about this in `phelp clrbar`. The position, `x` and `y`, are described as being between 0.0 and 1.0 with the lower left hand corner equaling 0;0.

REFVEC - Allows you to plot a reference arrow of specific size and location within your figure. The syntax is as follows: `refvec = Magnitude; x; y; text size/font/width/; string`.

STNPLT - Allows the user to plot station markers and station information.

We can set this parameter following this format: `text color / text attributes | marker attributes | stnfil.tbl # column`. This is a good thing to use if you are zoomed in on a small region. The `column` describes the an attribute of the station:

| | | | |
|---|---------|---|-----------|
| 1 | ID | | |
| 2 | Number | 6 | Latitude |
| 3 | Name | 7 | Longitude |
| 4 | State | 8 | Elevation |
| 5 | Country | 9 | Priority |

MARKER - This specifies the marker color, type, size, and line width separated by slashes. The marker types are listed here:

| | | | |
|----|--------------------|----|-----------------|
| 1 | plus sign | 13 | hourglass X |
| 2 | octagon | 14 | star |
| 3 | triangle | 15 | dot |
| 4 | box | 16 | large X |
| 5 | small X | 17 | filled octagon |
| 6 | diamond | 18 | filled triangle |
| 7 | up arrow | 19 | filled box |
| 8 | X with top bar | 20 | filled diamond |
| 9 | Z with bar | 21 | filled star |
| 10 | Y | 22 | minus sign |
| 11 | box with diagonals | 23 | tropical storm |
| 12 | asterisk | 24 | hurricane |

STREAM - *lines/arrows/stop/slow/scale*, syntax for streamlines.

GDTHGT

Draws contours and wind barbs or arrows on a time-height section plot at a point within a numerical model grid.

GDPOINT - Defines the point at which the time-height section is created.
Can be defined by the latitude;longitude, *@station_ID*, or graphically by using cursor *gpoint*

TAXIS - Contains the range, increment, and location for labels on a time axis input as: *start-stop-inc;lb;gl;tm*. Where *inc* is the time increment, *LB*, *GL*, and *TM* are the frequencies for labels, grid lines, and tick marks. To plot the time height section of a 48-hour forecast model, set `taxis = f00-f48`

GPCOLOR

Changes the colors on a color device. Set colors, then set the device (e.g. gif, ps, etc.), and then run it. Now when you use that device in another GEMPAK program, it will have the color attributes you set.

COLORS - Specifies a list of color numbers which must be separated using colons:
You can define the colors you list above for the device you specify in *gpcolor*. I would say the most useful thing in *gpcolor* is defining a background color (*color 101*). I would also encourage you to read the *phelp colors* for more information.

Here is a helpful hint, if you put something like this in a script (we'll learn about scripts in a couple weeks) in which you are making color plots, this will give you a white background:

```
gpcolor << EOF1
colors = 101 = 255:255:255
device = gif | image.gif | x;y sizes | C
r
e
EOF1
```

Here the *x;y sizes* are the number of pixels in the x and y direction of the image (not always necessary) and C corresponds to the device using color. If a G or M were in place of the C it would use a 20-count grayscale or a monochrome lines with color fills, respectively.

GDVINT

For some analyses, it may be useful to examine variables on a surface of constant potential temperature (given that air parcels conserve their potential temperature barring any diabatic effects). You may notice that the gridded data files we use do not have potential temperature as a vertical coordinate, which, at its face, may make this type of analysis substantially more difficult. Luckily, GEMPAK has a program that interpolates variables within one vertical coordinate (e.g. pressure) into another (e.g. theta). Here's an example of how to interpolate the 00-h forecast from today's 12Z NAM run from pressure to isentropic coordinates.

```
GDFILE = /weather/data/gemdata/hds/16092112_eta.gem
GDOUTF = 16092112_thta.gem (the name of the file GEMPAK will create)
GDATTIM = f00 (interpolate the 00-h forecast)
GVCORD = pres/thta (interpolate from pressure to theta coordinates)
GLEVEL = 275-390-5 (interpolate to levels every 5-K between 275 and 390 K)
MAXGRD = 29999 (you will typically never change this)
GAREA = 20;-125;55;-55 (same as in GDPLLOT)
VCOORD = pres (the vertical coordinate you are interpolating from)
```

Using isentropic analysis is particularly useful when examining processes from a Potential Vorticity (PV) perspective. We will get to this type of analysis much later on into the semester.

Just a note: Now you have the tools to create meaningful and presentable GEMPAK plots for your labs and future weather discussions. Remember, for the second discussion it will be a requirement to incorporate GEMPAK images. Feel free to ask for help if you need it.