AOS 452 - Lab 2
GEMPAK Part I – Surface and Sounding Programs

This lab will be the first of two labs aimed at (*re*)familiarizing you with GEMPAK (**GE**neral **M**eteorological **PA**c**K**age), a software package that includes several programs useful for decoding, plotting, and displaying meteorological data. Certain GEMPAK programs will work only with specific types of data or will perform only specific tasks. The first two letters of a GEMPAK program's name correspond with these data types or tasks:

SF—surface observation data
SN—sounding observation data
GD—gridded data
OA—objective analysis
GP—graphics capabilities

This lab will concentrate on working with observational datasets, so we will be using the SF- (**s**ur**f**ace) and SN- (**s**ou**n**ding) programs. The commonly used GEMPAK programs for plotting surface and sounding data are as follows:

`sfmap` – plot surface observation data (surface METARs, ship and buoy observations...)
`snmap` – create horizontal plots of upper air observations (using sounding data)
`snprof` – create vertical profiles of upper air observations (skew-T for example)
`sncross` – create cross sections using upper air observations

Data Sources

A variety of real-time weather data is accessible from the Synoptic Lab computers, including (but not limited to) surface and upper-air observations, radar and satellite imagery, and model output. Most of the data arrives in a coded form (one such format is called GRIB) via the LDM (Local Data Manager, managed by Pete Pokrandt). Once the data is received, it is converted into more useful formats that can be accessed by programs such as Weather and GEMPAK. Data in these useful formats is placed in the following directory:

`/weather/data/gemdata/`

The subdirectories below the `data/` directory hold data in different formats, including garp, gemdata, grib, mcidas, nexrad, and weather. In this lab, we will be using GEMPAK to analyze surface and upper-air observations.

Thus, we'll be interested in the `gemdata/` subdirectory, which contains files in GEMPAK format. Within `gemdata/` there are additional subdirectories, including `surface/` and `upperair/`. Use the `ls` command within the `gemdata/` directory to see the others. All of the data necessary for today's lab can be found in these directories.

Now `/labdata/data/gemdata/surface` is a lot of typing. Luckily, some shortcuts have already been set up for us. Simply use,

`$SAO` to refer to the surface data directory
`$UPA` to refer to the upper air data directory.

In UNIX-speak, these dollar-sign-prefixed names are called environmental variables. The `printenv` command displays every environmental variable currently set. Use the `ls` command on the `$SAO` and `$UPA` directories to see how the data files are named. The files we're interested in will have names of the form `yymmdd_sao.gem` (for surface obs) and `yymmdd_upa.gem` (for upper air obs).

The GEMPAK Basics

To begin using any of the GEMPAK programs, simply type the name of the program at a Unix prompt. All of the GEMPAK programs work similarly through the use of parameter lists. Once in one of the GEMPAK programs, it is useful to get a listing of all of the parameters and their settings by typing `l` (for 'list') followed by ENTER. To change a parameter, just use the following syntax:

parameter_name = value

GEMPAK has a useful help library for each parameter and each program. This library can be accessed with the **phelp** or **help** commands. (`phelp` displays the help one page at a time). As an example, if you were interested in the MAP parameter, you would type `phelp map`, while the command `phelp sfmap` would provide information and examples illustrating the use of the **sfmap** program. You must be within a GEMPAK program to use these commands.

Once you have changed all the parameters to your liking, type *r* for run from the prompt to create your plot.

Since GEMPAK has an extensive help library, these lab assignments will include only the essential information regarding the most frequently used parameters. If you need more information regarding parameters, use the *phelp* command, ask your friendly TA, or visit the GEMPAK documentation online at the following URL:

http://www.unidata.ucar.edu/packages/gempak/tutorial/manual/
(or google: gempak tutorial)

Remember that you must use gpend to close GEMPAK windows.

*NEVER click on the corner of an open GEMPAK window to close it. You will leave behind ghost GEMPAK processes that will continue to run, slowing down and eventually crashing your machine! You must also use **gpend** before printing newly created PostScript files.*

SFMAP – Surface Observations

As previously mentioned, `sfmap` is a GEMPAK program allowing one to create plots of surface (or ship/buoy) observation data. Here is a description of the most important parameters used in this program:

area:	The region in which to search for the observations
garea:	The viewing region
sfparm:	Variables to plot for each station and where to place the variables relative to the station's location
dattim:	Date and time to be used
sffile:	Surface observation file to be used
marker:	Symbol to be used to identify the station location
device:	Type of device (where to send the plot)

filter: Controls filtering of the data

You will find that *sfparm* does most of the action in the **sfmap** program. The values entered in *sfparm* determine what pieces of data are plotted. The order in which the variables are specified in *sfparm* determines where the variables are plotted. The order list and corresponding plotting location is as follows:

	8	
2	**10**	**4**
3	**1**	**5**
6	**11**	**7**
	9	

For example, if you want to have sky cover in position 1, temperature in Fahrenheit in position 3, and dew point temperature in Fahrenheit in position 6, you would enter the following:

```
sfparm = skyc;;tmpf;;;dwpf
```

(Note: Only the first 11 positions are shown on this lab for brevity. The newest version of **sfmap** has 25 positions available. See *phelp sfparm* in the **sfmap** program for more details.)

Drawing attributes of the *sfparm* parameters (such as size and width) also can be controlled. These attributes are specified following a colon after the variable name. For instance, the text for temperature can be made about half the size of the default text size by entering the following:

```
sfparm  =  tmpf:0.5
```

Check *phelp sfparm* for more details on adjusting the drawing attributes or other items related to *sfparm*.

Let us do an example that will give us output from the **sfmap** program. We will create a plot of surface observations from 1200 UTC 11 September 2011. Change the following parameters as outlined, and others that may be necessary:

```
area   = dset
garea  = TOP-- ← This represents Topeka KS zoomed out twice. Dash (-) is zoom out mechanism.
sfparm = skyc:0.75;tmpf;wsym;smsl;;dwpf;brbk:0.75
                 [Sky cover (size .75); Temperature (F); Weather Symbol; Abbreviated
                      Mean Sea Level Pressure; Dew Point; Wind barb in knots (size .75)]
sffile       = $SAO/110911_sao.gem
dattim       = 11/12
proj   = UTM
filter       = 1.0
map          = 8/1/2
text   = 0.75/3//hw
device       = xw
title  = 1/-3/Your Name Here
```

Once you are able to get the surface plot to appear in an x-window, modify the parameters to plot only the surface winds and sea level pressure.

SNMAP — Upper Air Observations

The GEMPAK program **snmap** can be used to create horizontal plots of upper air observations. The program is extremely similar to **sfmap**. However, some of the parameters are slightly different:

levels	Level on which to plot
snfile	Sounding data file to use
snparm	Same as *sfparm*, but with sounding variables
vcoord	Vertical coordinate used to define level plotting

Be sure you are exited out of **sfmap,** have typed **gpend**, and entered the **snmap** program.

SNPROF — Vertical Profiles

The **snprof** program plots profiles of upper air observation data. As you may have guessed, this means that **snprof** will use an upper air file (*snfile* = yymmdd_upa.gem) as in **snmap**. Here are a few parameters you will have to change to use **snprof**:

ptype:	Type of plot (e.g., skew-T, linear, log)
area:	Location over which the profile is drawn (use the @ symbol to identify a particular station; for example, *area* = @GRB)
stndex:	List of stability parameters to be plotted for the station
snparm:	Variables to be plotted
vcoord:	Vertical coordinate used to define level plotting
windpos:	Where to place the wind barbs on the plot
yaxis and *xaxis*:	Description of bounds on the axes, along with labeling frequency

Once in the **snprof** program, change the following parameters to get the vertical profile of temperature and dew point temperature in degrees Celsius from Minneapolis, MN at 1200 UTC 12 September 2011:

```
SNFILE   = $UPA/110912_upa.gem
DATTIM   = 12/12
AREA     = @mpx
SNPARM   = tmpc;dwpc
LINE     = 2;4/1;1/3;3/
PTYPE    = skewt
VCOORD   = pres
STNDEX   =
STNCOL   = 1
WIND     = bk1
WINPOS   = 1
MARKER   = 0
BORDER   = 1
TITLE    = 1/-2/Your name here
DEVICE   = xw
YAXIS    = 1000/100/100/;1;
XAXIS    = -40/40/10/;1;
```

```
FILTER   = 0.5
CLEAR    = yes
PANEL    = 0
TEXT     = 0.5/3//hw
THTALN   = 5/1/1/
THTELN   = 9/1/1/
MIXRLN   = 3/1/1/
```

After entering these parameters, type *run* to create the vertical profile for Minneapolis. Once you are able to complete this successfully, try getting a sounding for Green Bay, but from 1200 UTC on the 14th.

SNCROSS – Cross Sections

The GEMPAK program **sncross** can be used to draw cross sections through sounding data. Most of the parameters in **sncross** have already been described. The most important parameter in this program is *cxstns*, as it is used to define the line on which the cross section is taken. To define the cross section line in **sncross**, you list all of the upper air stations you wish to include separated by semicolons.

Create a cross section of 1200 UTC potential temperature on a line running from Medford, OR (MFR) to Arberdeen, SD (ABR). Set the following parameters:

```
CXSTNS   = mfr;boi;riw;rap;abr
 SNPARM   = thta
 SNFILE   = $UPA/110914_upa.gem
 DATTIM   = 14/12
 VCOORD   = pres
 PTYPE    = log
 YAXIS    = 1000/250/100/;1;
 TAXIS    =
 LINE     = 3/1/1/1
 BORDER   = 1
 CINT     = 3
 WIND     = 0
 TITLE    = 1/-3/Your name here
 PANEL    = 0
 DEVICE   = xw
 CLEAR    = y
 FILTER   = 0.5
 TEXT     = 0.5/3//hw
 CURVE    = 2
 CLRBAR   = 1/v/ll/0;0/.4/1|.6!1/v/ll/0;0/.4/1|.6
 CONTUR   = 3/1
 FINT     = 0
 FLINE    = 10-20
 CTYPE    = c
```

I strongly encourage you to take time on your own to become self-sufficient in using GEMPAK.

It will help you in your individual case studies immensely. The best way to learn GEMPAK is through trial and error. Remember to use gpend after exiting any GEMPAK program and before printing GEMPAK output! Future labs will become increasingly less descriptive with regard to basic GEMPAK commands.

PRINTING GEMPAK PLOTS

For most of this lab, we used *device = xw* to output plots to the computer monitor (an X-window). As the semester progresses, you will need paper copies of the plots you create in various GEMPAK programs. You can create files suitable for printouts by changing the device parameter. The file format most suitable for quality printouts is postscript. To write your output to a postscript file, use the following syntax:

device = ps|name_of_file.ps

Then, run the GEMPAK program like you normally do by typing *run* and pressing ENTER. You will not see a window display on your screen as you do when the device is set to *xw*. Rather, the output is being stored to a postscript file named in the *device* statement. After the GEMPAK program has created the file, exit the program, then type gpend.

A program called Ghostview allows you to view and print postscript files. To view the postscript file in Ghostview, type the following at the Unix prompt:

ggv name_of_file.ps &

A printout of the postscript file can be obtained in the Room 1411 printer by selecting "Print" under the File menu in the upper left corner of the window running Ghostview. After selecting "Print", a box will appear in which you will need to enter a print command or printer name (depending on what version of Ghostview is installed on the workstation). If asked for the print command, enter *lpr –P gpend*; if asked for the printer name, enter *gpend*. A printout can be obtained from the Room 1411 printer if you are not viewing the file in Ghostview by using the print command at the Unix prompt:

lpr -P gpend name_of_file.ps

Differences in color schemes between xw and postscript devices

Be aware that the X-window (xw) and postscript (ps) device parameters use two different color schemes. The only colors available in general postscript files are white, black, and varying shades of gray. The colors that you see in a plot in an X-window will not match the colors on a generic postscript file. A color scheme that may look strange on the computer monitor may work well in a postscript file. I suggest you check how your postscript file looks in Ghostview before printing a postscript file. The issue of color schemes will come more into play when we learn about color fills in an upcoming lab.