Introduction to GRIB2 using the GFS forecasts
2/2013 Wesley Ebisuzaki

GRIB2 (grib edition 2) is a standard format sponsored by the WMO (UN's World Meteorological Organization) for the transmission of gridded data between the national meteorological centers. In English, grib2 is a format for meteorological and oceanographic forecasts and analyses that is is used by the big guys that make the operational weather forecasts. If you want the model weather forecasts from a government meterological agency, the changes are that it will be available in grib format.

GRIB2 is a transmission format so compression is a high priority. Starting with a GFS forecast file, converting it to netcdf-3 increases the file size by 6.4 times.

```
-rwxr-xr-x 1 wd51we wd5  58043511 2013-02-08 09:55 gfs.t06z.pgrb2f12     (grib2)

-rw-r--r-- 1 wd51we wd5 372295888 2013-02-08 10:06 gfs.t06z.pgrb2f12.nc
  (netcdf3)
```

Grib2 is a transmission format but can be used by application programs and common database functions can be accomplished by utilities/programs.

GRIB2 is defined endian independent and and computer word size independent. However, 32-bit computer tend to limited to 2 GB files and grib2 software have been written assuming a word size of at least 32 bits.

What is in a GRIB2 file?

An easy way to see the contents of a grib2 files is to run wgrib2 on it. Wgrib2 is freely available for Windows, linux and Unix machines. Using a GFS forecast file,

```
bash-3.2$ wgrib2 gfs.t06z.pgrb2f12
1.1:0:d=2013020806:UGRD:planetary boundary layer:12 hour fcst:

1.2:0:d=2013020806:VGRD:planetary boundary layer:12 hour fcst:

2:232030:d=2013020806:VRATE:planetary boundary layer:12 hour fcst:

3:337239:d=2013020806:HGT:10 mb:12 hour fcst:

4:571416:d=2013020806:TMP:10 mb:12 hour fcst:

5:632506:d=2013020806:RH:10 mb:12 hour fcst:

6.1:647281:d=2013020806:UGRD:10 mb:12 hour fcst:

6.2:647281:d=2013020806:VGRD:10 mb:12 hour fcst:

...
312:56419187:d=2013020806:HGT:PV=-2e-06 (Km^2/kg/s) surface:12 hour fcst:

313:56671672:d=2013020806:PRES:PV=-2e-06 (Km^2/kg/s) surface:12 hour fcst:

314:56953195:d=2013020806:VWSH:PV=-2e-06 (Km^2/kg/s) surface:12 hour fcst:

315:57092940:d=2013020806:PRMSL:mean sea level:12 hour fcst:
```

```
316:57267394:d=2013020806:5WAVH:500 mb:12 hour fcst:

317:57345993:d=2013020806:GPA:1000 mb:12 hour fcst:

318:57710431:d=2013020806:GPA:500 mb:12 hour fcst:

319:57938644:d=2013020806:5WAVA:500 mb:12 hour fcst:
```

The format of the default inventory is

(message num)[.sub-message num]:(location):d=(reference time):(variable):(Z):(dtime)[:other information]

[...] are optional values
(...) are values as described by ...
underlined characters are literals

message num: message number =1..N messages contain a complete description of one or more fields
submesage num: if the message contains more than 1 field, then submess num = 1..number of fields in the message
location: the byte location (starting from zero) of the start of the message.
reference time: usually the analysis time or the start of the forecast time, the format is YYYYMMDDHH to see the minutes and seconds, use the -S option in wgrib2.
Variable: a short name of the variable. The variables names are not part of the WMO standard. Wgrib2 uses the NCEP names
Z: the vertical coordinate. It can be a level or a layer. Examples include 300 mb, 2 m above ground, atmospheric column, tropopause, 20 C oceanic isotherm and 10 m below the sea.
Dtime: a modifier of the reference time. Examples 12 hour forecast, 1 month average
other information: optional fields such as ensemble information, chemical species, probabilities. The "other information" is subject to more changes as grib2 format is augmented.

Interpretation

Here are lines 4 and 5 of the above inventory:

```
3:337239:d=2013020806:HGT:10 mb:12 hour fcst:

4:571416:d=2013020806:TMP:10 mb:12 hour fcst:
```

The first line says the message
  "3" grib message number 3
  "337239" message starts at byte location 337239 and ends at 571415
      571415 is one less than the start of the next message
 "d=2013020806" forecast started at 06Z Feb 8, 2013020805
  "HGT" the field is the geopotential height in geopotential meters (see NCEP tables)
  "10 mb" the field for 10 mb or 100 hPa.
  "12 hour fcst" the field is a 12 hour forecast from the intial time 2013020806

Messages and Sub-messages

Grib files consist of a sequence of grib messages. Each message is a complete description of one or more fields. This simple structure means that you can make grib files by joining two grib files.

Combining two grib files.

Linux: cat grib2 grib2 > grib3

A grib message is a complete description of a field. In the following inventory, the 10 mb HGT occupies byte locations 337239 to 571415 (234177 bytes).

```
3:337239:d=2013020806:HGT:10 mb:12 hour fcst:

4:571416:d=2013020806:TMP:10 mb:12 hour fcst:
```

You can extract byte locations 337239 to 571415 by using the dd command

```
bash-3.2$ dd if=gfs.t06z.pgrb2f12 ibs=1 skip=337239 count=234177 of=hgt.grb

...
bash-3.2$ wgrib2 hgt.grb

1:0:d=2013020806:HGT:10 mb:12 hour fcst:
```

Grib files can have submessages such as

```
6.1:647281:d=2013020806:UGRD:10 mb:12 hour fcst:

6.2:647281:d=2013020806:VGRD:10 mb:12 hour fcst:
```

Grib message 6 has two sub-messages. The first sub-message is the UGRD (zonal wind) and the second sub-message is VGRD (meridional wind). Sub-messages save space because they share a common grid definition. Submessages are often used to keep common variables together. For example, NCEP Operations usually keep the zonal and meridional winds in the same message as shown above. Note that in the above inventory, both fields start at the same byte location (647281).

Usually the grid defintion is on the order of 30 bytes, so using submessage only provides marginal savings. The only exception is for irregular grids where the latitude and longitude of each grid point is stored in the grid definition. In this case, the grid definition uses more space than the grid values. By using submessages, only one copy of the latitude longitude values need be saved.

Byte Location

Grib messages are like atoms. Each atom is compete and takes up a finite volume. If you had small fingers, you could grab that sodium atom and place it over there. A grib message has a starting byte location and an ending byte location.

Variables

Wgrib2 uses the NCEP variable names as given by http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2_doc.shtml
You can see the defintions by using the -v option.

```
bash-3.2$ wgrib2 -v gfs.t06z.pgrb2f12 | more

1.1:0:d=2013020806:UGRD U-Component of Wind [m/s]:planetary boundary layer:12 hour fcst:

1.2:0:d=2013020806:VGRD V-Component of Wind [m/s]:planetary boundary layer:12 hour fcst:

2:232030:d=2013020806:VRATE Ventilation Rate [m^2/s]:planetary boundary layer:12 hour fcst:

3:337239:d=2013020806:HGT Geopotential Height [gpm]:10 mb:12 hour fcst:

4:571416:d=2013020806:TMP Temperature [K]:10 mb:12 hour fcst:
```

Grib messages store 2 dimensional grids. Usually the grids are horizontal (x-y, lat-lon). For a 3 dimensional fields, you have a set of grib2 message with varying vertical levels/layers. Levels use one fixed surface and layers used two fixed surfaces.

http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2_table4-5.shtml

dtime

Column 3 of the inventory has the reference time which is typically the analysis or start of forecast tme (the time stamp of the intial conditions). In our GFS forecast, there are 3 different dtimes.

bash-3.2$ wgrib2 gfs.t06z.pgrb2f12 -match ":(TMP:2 m above ground|PRATE|APCP):"

195:35104909:d=2013020806:TMP:2 m above ground:12 hour fcst:

202:36504744:d=2013020806:PRATE:surface:6-12 hour ave fcst:

203:36620774:d=2013020806:APCP:surface:6-12 hour acc fcst:

The 2 m temperature is the 12 hour forecast
The precipitation rate (PRATE) is the average of the 6-12 hour forecast (6 hour average)
The accumulated precipitation (APCP) is the accumulation of the 6 to 12 hour forecast (6 hour accumulation)

Making smaller grib files

It is common to be overwhelmed by data volumes of the forecasts. NCEP could output higher resolution output files but the storage space and bandwidth need to distribute these files to the forecast offices and the public would break the budget. The end user also can have the same problem if they try to save the forecasts. One way to save space is to save only the important fields. Suppose you only wanted the APCP (accumulated precipitation) and the 2 m temperature.

```
bash-3.2$ wgrib2 gfs.t06z.pgrb2f12 -match ":(TMP:2 m above ground|APCP):" -\
   grib gfs.t06z.pgrb2f12.small

195:35104909:d=2013020806:TMP:2 m above ground:12 hour fcst:

203:36620774:d=2013020806:APCP:surface:6-12 hour acc fcst:


bash-3.2$ ls -l gfs.t06z.pgrb2f12*

-rwxr-xr-x 1 wd51we wd5 58043511 2013-02-08 09:55 gfs.t06z.pgrb2f12

-rw-r--r-- 1 wd51we wd5   264780 2013-02-08 15:31 gfs.t06z.pgrb2f12.small
```

By one command, the file has gone from 58 MB to 0.26 MB. Now suppose we are only interested in the region lon=-100 to -40 and lat=0 to 50. Turning on compression (-set_grib_type c3) and writing out a section of the grib file gives a 25 KB file.

```
bash-3.2$ wgrib2 gfs.t06z.pgrb2f12.small -set_grib_type c2 \
        -small_grib -120:-40 0:50 gfs.t06z.pgrb2f12.small.region

1:0:d=2013020806:TMP:2 m above ground:12 hour fcst:

2:191860:d=2013020806:APCP:surface:6-12 hour acc fcst:


bash-3.2$ ls -l *small*

-rw-r--r-- 1 wd51we wd5 264780 2013-02-08 15:31 gfs.t06z.pgrb2f12.small
```

```
-rw-r--r-- 1 wd51we wd5  24888 2013-02-08 15:45 gfs.t06z.pgrb2f12.small.region
```

To see the grid of the first grib message, use the -grid option and -d 1 to only view the first message.

```
bash-3.2$ wgrib2 gfs.t06z.pgrb2f12.small.region -grid -d 1

1:0:grid_template=0:winds(N/S):

        lat-lon grid:(161 x 101) units 1e-06 input WE:SN output WE:SN res 48

        lat 0.000000 to 50.000000 by 0.500000

        lon 240.000000 to 320.000000 by 0.500000 #points=16261
```

So in our example, the original GFS forecast was 58 MB. By selecting a few fields and restricting the domain, our hypothetical file is now svelte 25 KB. (Think boat.) Jpeg compression would produce a smaller file at hte cost of more cpu cycles to decode.