Solutions to grib_filter practicals 2018

Question 1

Run grib_filter with the rules files ‘print.filter’, ‘write.filter’, ‘transient.filter’ on ‘tigge.grib’

- Comment/uncomment the instructions one by one to see the different behaviours

Running grib_filter with the print.filter rules file and tigge.grib as the input GRIB file produces the following:

```
%  grib_filter print.filter tigge.grib
Hello ecCodes!
Hello ecCodes!
...
```

This is produced by the first line in the rules file:

```
print "Hello ecCodes!";
```

Note that this can also be done directly from standard input:

```
echo 'print "Hello ecCodes";' | grib_filter - tigge.grib
```

Next, comment the first line with a ‘#’ and uncomment the second line so that the active rule is:

```
print "- [count] - shortName=[shortName] date=[dataDate] step=[step][stepUnits:s]";
```

Running grib_filter with the new rules file produces the following output:

```
%  grib_filter print.filter tigge.grib
   - 1 - shortName=gh date=20180223 step=24h
   - 2 - shortName=q date=20180223 step=24h
   ...
   - 239 - shortName=tcw date=20180223 step=24h
   - 240 - shortName=tp date=20180223 step=24h
```

Using the third rule:

```
print ("out.txt") "- [count] - shortName=[shortName] date=[dataDate] step=[step][stepUnits:s]";
```

does not produce any output to the screen but instead writes the output to a file called out.txt.

The fourth line is:

```
print "[distinctLatitudes!1%.3f]";
```

Running with this rule prints the distinct latitudes array for each message as a single column with the numbers in decimal format with 3 decimal places:

```
%  grib_filter print.filter tigge.grib
  90.000
  88.500
  87.000
  85.500
  ...
```

The final rule is:

```
print "[latLonValues!3%.5f',']";
```

Running grib_filter with this rule prints the latitudes, longitudes and values for each message formatted as 3 comma-separated columns with the numbers in decimal format with 5 decimal places:

```
%  grib_filter print.filter tigge.grib
   90.00000,0.00000,1.00000,
   90.00000,1.50000,1.00000,
   90.00000,3.00000,1.00000,
   90.00000,4.50000,1.00000,
   ...
```

Running grib_filter with tigge.grib as the input GRIB file with the write.filter rules file:

```
write "[dataDate]_[centre].grib[edition]";
```

splits the messages in the input file into a separate files for each date, centre and GRIB edition combination:

```
%  grib_filter write.filter tigge.grib

%  ls *_*.grib*
  20180223_ammc.grib2  20180223_cwao.grib2  20180223_egrr.grib2  20180223_lfpw.grib2  20180223_rksl.
grib2
  20180223_babj.grib2  20180223_ecmf.grib2  20180223_kwbc.grib2  20180223_rjtd.grib2  20180223_sbsj.
grib2
```

Running with the second line of the rule uncommented:

```
write;
```

simply writes each message to the default output GRIB file, “filter.out”. This is identical to the input tigge.grib file:

```
% grib_compare tigge.grib filter.out

% echo $?
0
```

Running grib filter with the transient.filter rule:

```
transient mykey1 = (step == 4 ) * step;
transient mykey2 = (step == 24 ) * step;
print "-- [count] -- step=[step] mykey1=[mykey1] mykey2=[mykey2]";
```

Prints the following output to the screen:

```
-- 1 -- step=24 mykey1=0 mykey2=24
-- 2 -- step=24 mykey1=0 mykey2=24
-- 3 -- step=24 mykey1=0 mykey2=24
-- 4 -- step=24 mykey1=0 mykey2=24
-- 5 -- step=24 mykey1=0 mykey2=24
...
```

In the output seen here, mykey1 is set to 0 because step=24 and the first expression is not satisfied so (step == 4) = 0. But mykey2 is set to 24 because the second expression is satisfied and (step == 24) = 1. Effectively, " (step == 4)" acts a little like an "if": "If (step == 4) {mykey1 = step}".

Advanced grib_filter practicals

Question 1

Change the date to 20170301 and the step to step+48 in the file ‘tigge.grib’ only for the data produced by ECMWF.

The filter rule file is:

```
#
# Question 1 solution
#
# $> grib_filter question1.filter tigge.grib
#

if (centre is "ecmf" ) {
  set dataDate = 20170301;
  set step = step + 48;
}

write "question1.grib";
```

Note that the test on centre as a string uses "is" - `if (centre is "ecmf")`.

Question 2

Set the values of the first message in the file ‘tigge.grib’ to 1.2, 3.4, 5.6, 3.7 and step to 72. Write only this message to the file ‘question2.grib’

- Check the values coded with grib_get_data or grib_dump.

The filter rule file is:

```
#
# Question 2 solution
#
# $> grib_filter question2.filter tigge.grib
#

if (count == 1) {
    set step = 72;
    set values = {1.2,3.4,5.6,3.7};
    write "question2.grib";
}
```

Note that we identify the first message as `count == 1`.

We can check the result of running the filter by using grib_get_data:

```
% grib_filter question2.filter tigge.grib

% grib_get_data -P step -F"%.2f" question2.grib
Latitude, Longitude, Value, step
   90.000    0.000 1.20 72
   90.000    1.500 3.40 72
   90.000    3.000 5.60 72
   90.000    4.500 3.70 72
```

If you choose to set stepRange instead then note that this key is a 'string' so you need use:

```
 set stepRange = "72" ;
```

Question 3

Append to ‘question2.grib’ all the messages containing the same parameter of the other centres that are not encoded using a reduced Gaussian grid, setting the step to 72.

The filter rule file is:

```
#
# Question 3 solution
#
# $> grib_filter question3.filter tigge.grib
#

if (count != 1 &&
   shortName is "gh" &&
   !(gridType is "reduced_gg" )) {
    set step = 72;
    append "question2.grib";
}
```

Note the construct for "NOT IS". The "is" test is negated with the "!":

```
!(gridType is "reduced_gg" )
```

Check the output with grib_ls

```
        % grib_ls question2.grib
        question2.grib
        edition      centre       date         dataType     gridType     stepRange    typeOfLevel
level        shortName    packingType
        2            ammc         20180223     pf           regular_ll   72           isobaricInhPa
1000         gh           grid_simple
        2            babj         20180223     pf           regular_ll   72           isobaricInhPa
1000         gh           grid_simple
        2            cwao         20180223     pf           regular_ll   72           isobaricInhPa
1000         gh           grid_simple
        2            egrr         20180223     pf           regular_ll   72           isobaricInhPa
1000         gh           grid_simple
        2            kwbc         20180223     pf           regular_ll   72           isobaricInhPa
1000         gh           grid_simple
        2            lfpw         20180223     pf           regular_ll   72           isobaricInhPa
1000         gh           grid_simple
        2            rjtd         20180223     pf           regular_ll   72           isobaricInhPa
1000         gh           grid_simple
        2            rksl         20180223     pf           regular_ll   72           isobaricInhPa
1000         gh           grid_simple
        2            sbsj         20180223     pf           regular_gg   72           isobaricInhPa
1000         gh           grid_simple
        9 of 9 grib messages in question2.grib

        9 of 9 total grib messages in 1 files
```

Question 4

Split ‘tigge.grib’ into several files, one for each centre, containing only surface parameters and parameters that are at level 10 of height above ground.

- For the surface parameters, set changeDecimalPrecision to 2,
- For the height above ground parameters set changeDecimalPrecision to 3.
- Print information messages for each case, such as:

```
Centre ammc parameter v not written
Centre ammc parameter 10u written to question4-ammc.out
```

The filter rule is:

```
#
# Question 4 solution
#
# $> grib_filter question4.filter tigge.grib
#

if ((typeOfLevel is "surface") || ((typeOfLevel is "heightAboveGround") && (level == 10))) {

    switch (typeOfLevel) {

        case "surface":
            set changeDecimalPrecision=2;
        case "heightAboveGround":
            set changeDecimalPrecision=3;
        default:
            print "Unknown level type!";
    }

    print "Centre [centre] parameter [shortName] written to question4-[centre].grib";
    write "question4-[centre].grib";

} else {

   print "Centre [centre] parameter [shortName] not written";

}
```

Question 5

Merge the messages from the previously split GRIB files into a single file

- Write only messages encoded in a regular lat-long grid, and exclude messages where the parameters are 10u or 10v.The

The filter rule is:

```
#
# Question 5 solution
#
# $> grib_filter question5.filter question4-*.grib
#

if ((gridType is "regular_ll") && !(shortName is "10u") && !(shortName is "10v") ) {
    write "question5.grib";
}
```