Solutions for GRIB tools extra modify practicals 2018

Question 1

Use grib_copy to split file1.grib1 into separate files for each parameter and level combination

To split the file tz_an_pl.grib1 into separate files for each parameter/pressure level combination, use:

```
%  grib_copy tz_an_pl.grib1 "[shortName]_[level].grib[edition]"
```

```
%  ls ?_*.grib1
t_1000.grib1 t_400.grib1 t_700.grib1 z_1000.grib1 z_400.grib1 z_700.grib1
t_300.grib1  t_500.grib1 t_850.grib1 z_300.grib1  z_500.grib1 z_850.grib1
```

Each file contains one message only, e.g.:

```
% grib_ls  -p centre,shortName,typeOfLevel,level,dataDate t_1000.grib1
t_1000.grib1
centre       shortName    typeOfLevel  level        dataDate
ecmf         t            isobaricInhPa  1000         20180223
1 of 1 grib messages in t_1000.grib1

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

Question 2

An SST field has been created by masking the Soil Temperature at Level 1 (STL1) with the Land-Sea Mask and is included with other messages in the file surface.grib1

- Use grib_set to change the parameter for the field from STL1 to SST and level type to ‘surface’.
- Be careful not to change the other parameters !
- Repeat with each different message output to a separate file.

To change the parameter from STL1 to SST and set the typeOfLevel to surface use:

```
% grib_set -w shortName=stl1 -s shortName=sst,typeOfLevel=surface surface.grib surface_new.grib1
```

[Alternatively, you could set paramId=34, etc.]

Make sure that only the STL1 parameter is changed by using the -w option.

Checking the new file with grib_ls shows:

```
% grib_ls -p centre,shortName,levelType,typeOfLevel,level,dataDate,dataTime,stepUnits:s,stepRange
surface_new.grib1

  surface_new.grib1

centre       shortName    levelType    typeOfLevel  level        dataDate     dataTime     stepUnits    step
Range

ecmf         lsm          sfc          surface      0            20180223     0            h            0

ecmf         sst          sfc          surface      0            20180223     0            h            6


ecmf         msl          sfc          surface      0            20180223     0            h            6


ecmf         sst          sfc          surface      0            20180223     0            h
12

ecmf         msl          sfc          surface      0            20180223     0            h            12

ecmf         sst          sfc          surface      0            20180223     0            h            18

ecmf         msl          sfc          surface      0            20180223     0            h            18

ecmf         sst          sfc          surface      0            20180223     0            h            24

ecmf         msl          sfc          surface      0            20180223     0            h            24
9 of 9 grib messages in surface_new.grib1

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

To output each message to a separate file, use:

```
% grib_set -w shortName=stl1 -s shortName=sst,typeOfLevel=surface surface.grib1 "[shortName]_
[stepRange].grib1"
```

This creates 9 separate files named, e.g., sst_6.grib, each containing a single GRIB message.

It is also possible to use the message number:

```
% grib_set -w shortName=stl1 -s shortName=sst,typeOfLevel=surface surface.grib "surface_[count].grib1"
```

Question 3

Use grib_to_netcdf to convert the data in file4.grib1 to NetCDF

- What happens ?
- Follow the hint and try again !
- Inspect the content with ncdump

Using grib_to_netcdf to convert file4.grib1 to NetCDF gives an error:

```
% grib_to_netcdf -o out4.nc file4.grib1
grib_to_netcdf: Version 2.6.0
grib_to_netcdf: Processing input file 'file4.grib1'.
ECCODES ERROR   :  Wrong number of fields
ECCODES ERROR   :  File contains 8 GRIBs, 8 left in internal description, 6 in request
ECCODES ERROR   :  The fields are not considered distinct!

ECCODES ERROR   :  Hint: This may be due to several fields having the same validity time.
ECCODES ERROR   :  Try using the -T option (Do not use time of validity)
```

The conversion fails because grib_to_netcdf tries to set the 'time' variable based on the validity date and time of the GRIB data. In this case, more than one message has the same validity date and time, for example, message number 3 and message number 5 (1800 on 20180223) and message number 4 and message number 6 (0000 on 20180224):

```
% grib_ls -n time -P count file4.grib1
file4.grib1

count         dataDate      dataTime      stepUnits     stepType      stepRange     startStep     endStep
    validityDate  validityTime

1             20180223      0000          h             instant       6             6             6
    20180223      600

2             20180223      0000          h             instant       12            12            12
    20180223      1200

3             20180223      0000          h             instant       18            18            18
    20180223      1800

4             20180223      0000          h             instant       24            24            24
    20180224      0

5             20180223      1200          h             instant       6             6             6
    20180223      1800

6             20180223      1200          h             instant       12            12            12
    20180224      0
```

```
7             20180223      1200          h             instant       18            18            18
    20180224      600

8             20180223      1200          h             instant       24            24            24
    20180224      1200

    8 of 8 messages in file4.grib1

    8 of 8 total messages in 1 files
```

Follow the hint and re-run the grib_to_netcdf command with the "-T" option:

```
        % grib_to_netcdf -T -o out4.nc file4.grib1
        grib_to_netcdf: Version 2.6.0
    grib_to_netcdf: Processing input file 'file4.grib1'.
    grib_to_netcdf: Found 8 GRIB fields in 1 file.
    grib_to_netcdf: Ignoring key(s): method, type, stream, refdate, hdate
    grib_to_netcdf: Creating netCDF file 'out4.nc'
    grib_to_netcdf: NetCDF library version: 4.4.1 of Aug  3 2016 11:10:49 $
    grib_to_netcdf: Creating large (64 bit) file format.
    grib_to_netcdf: Defining variable 't2m'.
    grib_to_netcdf: Done.
```

Now the conversion is successful. The NetCDF file now contains two variables to describe the time - time and step:

```
        % ncdump -v time,step  out4.nc

        netcdf out4 {

        dimensions:

                longitude = 2 ;

                latitude = 2 ;

                step = 4 ;

                date = 2 ;

        variables:

                        ...

                int step(step) ;

                        step:units = "hours" ;

                        step:long_name = "time_step" ;

                int date(date) ;

                        date:units = "days since 1900-01-01 00:00:0.0" ;

                        date:long_name = "Base_date" ;

                        ...

        data:

           step = 6, 12, 18, 24 ;


      time = 0, 12 ;
    }
```

If the data covers more than one date then there will be a third variable - date - to describe the time.