How to convert GRIB to CSV

Created by Xiaobo Yang, last modified by Karl Hennermann on Aug 01, 2018

You might find it easier to handle data in a CSV file instead of a GRIB file. This article demonstrates how to extract data from a GRIB file using ECMWF's ecCodes and save the output as a CSV file. You are expected to have ecCodes installed on a Linux machine before you continue. You are also recommended to add ecCodes' **bin** directory to your **PATH**. Work below was tested with ecCodes 2.5.0.

Step-by-step guide

1. Download 2m temperature from the Copernicus Climate Change Service (C3S)'s Climate Data Store (CDS) and save the data as file **t2m_20000801.grib**. In this example, selections are set as below:
    a. Dataset: ERA5 hourly data on single levels from 2000 to 2017 (you may see different years as new data is released)
    b. Parameter : 2m temperature
    c. Product type: Reanalysis
    d. Year : 2000
    e. Month : August
    f. Day : 01
    g. Time : 06:00, 12:00 and 18:00
    h. Format : GRIB
2. Check the GRIB file by issuing command **grib_ls -P time t2m_20000801.grib**, you should see three GRIB messages in the file at three different times: 600, 1200, 1800 as shown below. Remember three time values were selected in step 1. Option **-P time** shows key "time", typing **grib_ls** for help.

```
t2m_20000801.grib
time         edition      centre       typeOfLevel  level        dataDate     stepRange
600          1            ecmf         surface      0            20000801     0
1200         1            ecmf         surface      0            20000801     0
1800         1            ecmf         surface      0            20000801     0
3 of 3 messages in t2m_20000801.grib

3 of 3 total messages in 1 files
```

3. Say I want to extract **lat, lon, 2t** (2m temperature) at time = 12:00 from the GRIB file. Run command **grib_get_data -w time=1200 t2m_20000801.grib > temp.csv** and you will get a file containing data like below. Option **-w time=1200** filters data for time = 12:00 only, typing **grib_get_data** for help.

```
Latitude, Longitude, Value
   90.000    0.000 2.7346786499e+02
   90.000    0.250 2.7346786499e+02
   90.000    0.500 2.7346786499e+02
   90.000    0.750 2.7346786499e+02
...
```

4. Format the CSV file. You may use script below.

```
#!/usr/bin/env python
```

```
"""
Save as format.py, then run "python format.py".

Input file : temp.csv
Output file: t2m_20000801.csv
"""
with open('temp.csv', 'r') as f_in, open('t2m_20000801.csv', 'w') as f_out:
    f_out.write(next(f_in))
    [f_out.write(','.join(line.split()) + '\n') for line in f_in]
```

5. You have a CSV file **t2m_20000801.csv**, which is ready to be imported in Excel. Notice there are over 1 million records in the file! You may now want to extract data for time=600 and 1800.

```
Latitude, Longitude, Value
90.000,0.000,2.7346786499e+02
90.000,0.250,2.7346786499e+02
90.000,0.500,2.7346786499e+02
90.000,0.750,2.7346786499e+02
...
```

If you have CDS API and ecCodes (with its Python interface) installed, you will be able to retrieve (in GRIB), extract and export data to CSV by writing up a Python script.

Related articles

- ERA5: What is the spatial reference
- Extract data from GRIB/NetCDF for a specific location and time
- How to convert GRIB to CSV
- ERA-Interim: What is the spatial reference
- Details of ERA-Interim parameters

grib kb-how-to-article csv