

COSMO-REA6 Starthilfe

(zur Verwendung unter UNIX/Linux)

Das folgende Dokument soll die Handhabung der COSMO-REA6 Felder erleichtern. Im Originalformat liefert COSMO-REA6 stündliche Felder im grib-Format (DWD grib1).

Um die Datenmenge handhabbar zu machen, sind die einzelnen Parameter wie z.B. Windkomponenten U und V, Temperatur usw., jeweils in einzelne Files gepackt. Je nach 2D oder 3D Parameter sind diese monatlich bzw. täglich gepackt, und mit bzip2 komprimiert. Diese Dateien sind ca. 1GB bzw. 0.1GB groß. Nach dem Entkomprimieren liegen die Dateien wieder in *.grib vor und können mithilfe von `wgrib`, `eccodes` oder `CDO` gemäß den nachfolgenden Befehlen bearbeitet werden.

Informationen über konstante Felder wie Höhen der Modelllevel, Land-See Verteilung oder Bodentypen, sowie die Koordinaten der Gitterpunkte sind in der Datei [COSMO_REA6_CONST_withOUTsponge.grib](#) enthalten. COSMO-REA6 wird auf einem rotierten Längengrad-Breitengrad-Gitter gerechnet. Für einen Vergleich der Reanalysedaten mit z.B. Stationsbeobachtungen sind die nicht rotierten Koordinaten **RLAT** und **RLON** im geographischen System aus der konstanten grib- (oder nc-) Datei zu verwenden. Ebenfalls im konstanten File enthalten ist die Variable HHL, welche die Höhen der 41 Modellnebenflächen über Meeresebene definiert. 3D-Variablen werden an den Modellhauptflächen, den Schichtmitten, ausgegeben. Um die entsprechende Geberhöhe der Modellparameter über Meeresebene zu ermitteln ist ein arithmetischer Mittelwert der Höhen der begrenzenden Modellnebenflächen nötig. Zum Beispiel ist die unterste Modellschicht durch HHL[41] und HHL[40] begrenzt, das zugehörige Modelllevel liegt über dem Meer auf 10m Höhe. Die hier abgegebenen 6 untersten Modelllevel liegen über dem Meer in einer Höhe von 10m, 35m, 69m, 116m, 178m und 258m. Über dem Land variiert die Höhe der Modelllevel mit der Topographie.

Weitere Erklärungen zum COSMO-Gitter und den Modelllevel finden sich in http://www.cosmo-model.org/content/model/documentation/core/cosmo_userguide_5.04.pdf https://www.dwd.de/SharedDocs/downloads/DE/modelldokumentationen/nwv/cosmo_eu/cosmo_eu_dbbeschr_201406.pdf?__blob=publicationFile&v=3

CDOs (Climate Data Operators)

Die CDOs bieten eine Vielzahl an Operatoren zur Auswahl, Modifikation und statistischen Berechnung von grib und netcdf Dateien.

Download: <https://code.mpimet.mpg.de/projects/cdo/files>

Dokumentation: https://code.mpimet.mpg.de/projects/cdo/embedded/cdo_refcard.pdf

Informationen erhalten

Es können Informationen über das verwendete Gitter oder die Variablen erhalten werden:

```
cdo griddes infile
```

```
cdo sinfon infile
```

Ausschneiden eines kleineren Modellgebiets

Mit den Eckpunkten Südwest(lon1,lat1) und Nordost(lon2,lat2):

```
cdo sellonlatbox,lon1,lon2,lat1,lat2 infile outfile
```

Berechnung der Windgeschwindigkeit

Die Windgeschwindigkeit kann aus den Horizontalkomponenten U und V berechnet werden. Dabei kann die skalare Variable Windgeschwindigkeit sowohl auf dem rotierten als auch auf dem geographischen Gitter berechnet werden.

```
cdo chname,U,ws -sqrt -add -sqr -selname,U infile1 -sqr -  
selname,V infile2 outfile
```

Auswählen bestimmter Zeitschritte

Hier werden die Tage 3, 4, 5 der Monatsdatei ausgewählt:

```
cdo selday,3,4,5 infile outfile
```

Konvertierung von grib nach netcdf unter Berücksichtigung des Gitters

Die Voraussetzung jeglicher Konvertierung ist, dass die Metadatenvariable *timeRangeIndicator* auf null gesetzt ist, weil die CDOs ansonsten weitere Metadaten falsch interpretieren. Für die meisten Parameter (instantane Größen) ist das bereits der Fall, für gemittelte, akkumulierte und MinMax Parameter muss der *timeRangeIndicator* aktiv auf null gesetzt werden:

```
grib_set -s timeRangeIndicator=0 infile.grb outfile.grb
```

Bei der anschließenden Konvertierung gibt es zwei prinzipielle Vorgehensweisen.

1: Projizierung auf ein reguläres Lon-Lat Gitter

Hier wird die originale REA6 Datei mit Hilfe einer benutzerspezifischen Gitterdefinition in ein reguläres Gitter projiziert. Das grib-Format wird beibehalten:

```
cdo remapcon,outgrid.txt infile.grb outfile.grb
```

Im folgenden Beispiel wird zusätzlich ins netCDF-3-Format konvertiert:

```
cdo -f nc remapcon,outgrid.txt infile.grb outfile.nc
```

Es ist auch möglich, ins netCDF-4-Format zu konvertieren, das zusätzlich die Daten komprimiert und in etwa eine gleichwertige Speicherreduzierung wie das grib-Format erreicht:

```
cdo -f nc4 -z zip=9 -remapcon,outgrid.txt infile.grb outfile.nc4
```

Hierbei bezeichnet `outgrid.txt` eine Textdatei, in der die Eigenschaften des neuen Gitters beschrieben werden. Unteres Beispiel zeigt ein nicht rotiertes Längengrad-Breitengrad-Gitter mit 500x500 Gitterzellen, einer Auflösung von 12km und einem südwestlichen Eckpunkt bei 30°W und 30°N:

Outgrid.txt:

```
gridtype = lonlat
xsize =500
ysize = 500
xfirst = -30
yfirst = 30
xinc = 0.11
yinc = 0.11
```

2a: Transformation auf das originale zurück gedrehte COSMO Gitter: skalare Variablen

Hierfür wird die Datei [griddes_REA6.txt](#) verwendet, um die originale COSMO Gitterstruktur beizubehalten. Die Koordinaten werden von dem nativen COSMO Gitter (Pol rotiert) auf geographische Koordinaten geändert, das heißt, von regulären $X(\text{lon}(x), \text{lat}(y))$ auf irreguläre $X(\text{lon}_1(x,y), \text{lat}_1(x,y))$ mit $x=1,..,848$ and $y=1,..,824$. Für skalare Variablen stimmen die neuen Längen- und Breitenfelder mit den geograpstaghischen Orten überein, wohingegen die alten Längen- und Breitenfelder nur auf dem rotierten Gitter gelten.

```
cdo -f nc4 -z zip=9 -copy -setgrid,griddes_REA6.txt in.grb out.nc4
```

2b: Transformation auf das originale zurück gedrehte COSMO Gitter: vektorielle Variablen

Bei vektoriellen Variablen wie beispielsweise dem bodennahen Wind (U_{10M}, V_{10M}) erfordert die Rück-Rotation eine Darstellung der Komponenten U_{10M} und V_{10M} in dem neuen Koordinatensystem. Die Datei `in.grb` enthält beide Komponenten:

```
cdo -rotuvb,var33,var34 in.grb rotuvb.grb
```

Bei den auf Modelllevel vorliegenden Windkomponenten (U, V) muss zusätzlich das versetzte Gitter berücksichtigt werden:

```
cdo -rotuvb,var33,var34 -uvDestag,var33,var34 in.grb rotuvb.grb
```

Die Umwandlung auf das geographische Gitter kann anschließend wie oben beschrieben durchgeführt werden:

```
cdo -f nc4 -z zip=9 -copy -setgrid,griddes_REA6.txt rotuvb.grb
out.nc4
```

Die Datei `out.nc4` ist im Prinzip fertig, aber viele Metadaten stimmen noch nicht. Diese müssen korrigiert werden, da die CDOs mit der obigen Anweisung in den meisten Fällen eine zusätzliche Dimension sowie zusätzliche Variablen *height* und *grid_mapping_1* hinzufügen. Außerdem ist der Name der Variablen falsch gesetzt.

Die Folgenden Befehle basieren auf Kommandos, die im Paket NCO (source code und Dokumentation: <http://nco.sourceforge.net/>) bereitgestellt werden.

Zuerst wird der generische Variablenname geändert. In diesem Beispiel lautet dieser *var11* und der korrigierte Variablenname *T_2M*; **bitte entsprechend anpassen**:

```
ncrename -h -O -v var11,T_2M out.nc4
```

Der nächste Befehl löscht die Variable *grid_mapping_1*:

```
ncks -C -h -O -x -v grid_mapping_1 out.nc4 out.nc4
```

Dieser Befehl editiert die Attribute der Datenvariable, **bitte entsprechend anpassen**:

```
ncatted -h -O -a table,T_2M,d,,
        -h -O -a grid_mapping,T_2M,d,,
        -h -O -a coordinates,T_2M,d,,
        -h -O -a standard_name,T_2M,c,c,"air_temperature"
        -h -O -a long_name,T_2M,c,c,"2m air temperature"
        -h -O -a units,T_2M,c,c,"K" out.nc4
```

In vielen Fällen fügt CDO eine Dimension und Variable *height* hinzu. Das kommt wahrscheinlich dadurch zustande, dass die Variable *level* in den Metadaten der grib-Datei auf ungleich Null gesetzt ist. Die beiden folgenden Befehle löschen die Dimension und Variable *height*. **Beachte**: um die Dimension zu löschen, muss die gesamte Datei in den Arbeitsspeicher geladen werden, so dass viel Speicher zur Verfügung stehen muss:

```
ncwa -a height out.nc4 new.nc4
ncks -C -h -O -x -v height new.nc4 out.nc4
rm -fv new.nc4
```

Der letzte Schritt besteht darin, die globale Variable *history* zu korrigieren, **entsprechend den Namen der Datenvariablen anpassen**:

```
DATE=$(LANG=en_us_88591; date)
ncatted -h -O -a history,global,d,,
        -O -a history,global,c,c,"$DATE: COSMO-REA6 2m
temperature data converted to netCDF" out.nc4
```

Berechnung der Windrichtung

Die Windrichtung selber ist zwar eine skalare Größe, hängt aber von den Windrichtungsvektoren ab. Deswegen muss die Windrichtung auf dem Gitter berechnet werden, auf dem es gelten soll. Normalerweise interessiert die Windrichtung auf dem geographischen Gitter, also müssen wie im vorigen Absatz beschrieben zuerst die Windkomponenten U und V auf dieses Gitter rotiert werden, da alle Windkomponenten in REA6 auf dem rotierten Gitter bereitgestellt werden. Es sei hier nochmals darauf hingewiesen, dass die 2D Variablen U_10M und V_10M bereits *destaggered sind*, d.h. die Versetzung der Variablen gegenüber dem geographischen Gitter wurde bereits herausgerechnet. Zur Berechnung der Windrichtung aus den Windkomponenten kann danach folgende Befehlsabfolge ausgeführt werden:

```
cdo atan2 U_10M V_10M winddir
cdo -divc,3.14159265359 winddir o
cdo mulc,180. o winddir
cdo addc,180. winddir o
cdo setvar,winddir o winddir
rm -f o
```

Grib Api und ecCodes

ecCodes ist ein Softwaretool entwickelt am EZMW (Europäisches Zentrum für Mittelfristwettervorhersage). Es gilt als Weiterentwicklung von Grib Api und dient der Dekodierung von Dateien in grib und bufr Format.

Download und Dokumentation:

<https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home>

Beispiele:

- Umwandeln von grib in netcdf Datei:

```
grib_to_netcdf -o output.nc input.grib
```

- Auflisten der Dateiinhalte und Zusatzinformationen wie Datum, Level, Variablenname, Gittertyp, Wert, etc.

```
grib_ls input.grb
```

- Nächsten Punkt zu longitude/latitude (hier 50.5N, 12.34E) identifizieren: Dies funktioniert nur für nicht rotierte Gitter. Bei Anwendung auf COSMO-REA6 Felder muss somit vorher die oben angesprochene Projizierung mittels CDO auf ein regelmäßiges nicht rotiertes lonlat Gitter durchgeführt werden.

```
grib_ls -l 50.5,12.34,1 input.grb
```

- Auswahl der ersten drei Zeitschritte und Abspeicherung in neuer Datei:

```
grib_copy -w count=1/2/3 input.grb output.grb
```

Wgrib

Wgrib ist eine veraltete Software. Da sie aber zum Teil immer noch verwendet wird, wird sie hier kurz erwähnt.

Download und Dokumentation: <http://www.cpc.ncep.noaa.gov/products/wesley/wgrib.html>

Beispiele:

- Kurze Auflistung des Dateiinhalts (Variablenname und record number):

```
wgrib -s gribfile
```

- Konvertierung der gesamten Datei in ASCII Format:

```
wgrib -d all -text -o outfile.txt gribfile
```

- Auswahl einer bestimmten Variablen, hier Nummer 20:

```
wgrib -V -d 20 gribfile
```

Python

Bei Weiterverarbeitung der COSMO-REA6 Daten mit Python kann folgende library zur Verarbeitung von grib Dateien eingebunden werden:

```
import pygrib
```

<https://pypi.python.org/pypi/pygrib>

R

Bei Weiterverarbeitung mit R bietet sich die Umwandlung in netcdf Dateien an, da diese mit den entsprechenden Bibliotheken eingelesen und verarbeitet werden können:

```
library(ncdf4)
```

oder

```
library(RNetCDF)
```

Zum direkten Einlesen von grib (1 und 2) Dateien bietet sich insbesondere das Paket `gribr` an:

```
Library(gribr)
```

Dieses Paket ist allerdings nicht unter CRAN verfügbar, sondern muss von GITHUB herunter geladen werden und dann selbständig installiert werden. Die technische Voraussetzung ist eine Installation der ecCodes (siehe oben), da das Paket auf der Funktionalität dieser Software aufbaut.

Es gibt auch weitere Pakete, die grib Dateien einlesen und verarbeiten können. Dazu zählt beispielsweise das folgende Paket:

```
library(rgdal)
```

Matlab

Download und Dokumentation von Routinen zur Weiterverarbeitung mit Matlab netcdf/grib reader:

<http://www.mathworks.com/matlabcentral/fileexchange/21579-netcdf-grib-reader>

IDL

Download und Dokumentation von Routinen zur Weiterverarbeitung mit IDL:

http://www.harrisgeospatial.com/docs/GRIB_Routines.html