

Konvertierung der REA6 Daten

Vorgehensweise bei der Konvertierung von grb nach netCDF

Dieser Leitfaden wird zusätzlich zu unserem Dokument [COSMO REA6 Starthilfe.pdf](#) angeboten. Hier wird erklärt wie die originalen REA6 grb Dateien nach netCDF4 konvertiert werden. Dafür wird die Datei [griddes_REA6.txt](#) benötigt, um das originale Gitter beizubehalten. Die Koordinaten werden in einem unstrukturierten geographischen Längen-Breiten-Feld beibehalten. Zusätzlich werden die Metadaten der netCDF Datei korrigiert.

Die Voraussetzung dafür, dass diese Anleitung funktioniert ist, dass eine funktionierende Installation von CDO und NCO vorliegt.

Für instantane Daten (timeRangeIndicator=0) gilt diese Anleitung direkt. Der timeRangeIndicator der originalen grb Dateien für gemittelte, akkumulierte und min/max Daten muss zuerst gleich Null gesetzt werden:

```
grib_set -s timeRangeIndicator=0 in.grb out.grb
```

Der folgende Befehl führt die Konvertierung von grb nach netCDF4 mit hoher Komprimierung durch indem er die [Datei der Gitterbeschreibung](#) verwendet:

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

Im Prinzip ist die Konvertierung damit durchgeführt und man kann mit der Datei arbeiten. Allerdings fügt CDO in vielen Fällen eine zusätzliche Dimension hinzu, welche die Höhe der Daten beschreibt. Auch werden Variablen hinzugefügt, die *height* und *grid_mapping_1* heißen. Außerdem ist der Name der Datenvariable falsch kodiert worden. Die folgenden Befehle räumen die Metadaten der netCDF Datei auf löschen die zusätzlichen Dimensionen. Eine weitere Voraussetzung ist, dass NCO bis zu 4GM RAM allokiert sein kann.

Dieser Befehl benennt den generischen Variablennamen um. In diesem Beispiel lautet er *var11* und der korrigierte *T_2M*; **bitte entsprechend anpassen**:

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

Dieser 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
```

Die beiden folgenden Befehle editieren die globale *history* Variable, **auch hier 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
```

In vielen Fällen fügt CDO eine Dimension und Variable *height* hinzu. Das kommt wahrscheinlich dadurch zustande, wenn 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 bestehe wieder darin, die globale Variable *history* zu korrigieren, **auch hier 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
```