

Vom Server zum gridfile

Der Weg der MOPITT-Daten hinein in
die CO-Randwerte-Datei

27.01.2012 ECMWF-Datensätze auf Modellniveaus können genutzt werden

1 Holen der Rohdaten mit get_mopitt.ksh

Ich benutze für das Holen das Skript `get_mopitt.ksh`:

```
#!/usr/bin/ksh
USER=anonymous
PASSWORD=r.walter@fz-juelich.de
SERVER=ftp://14ft101.larc.nasa.gov
DIRECTORY=/private.scratch/icg181/mopitt_src
for k in 2001 2002 2003 2004 2005 2006 2007 2008
do
  for (( m=1; m<13;m++ ))
  do
    if (( m < 10 ))
      then i=0$m
    else i=$m
    fi

    for ((d=1;d<32;d++))
    do

      if (( d < 10 )) ; then j=0$d ; else j=$d ; fi

      wget -nc -v --directory-prefix=${DIRECTORY} \
      --user=$USER --password=${PASSWORD} \
      ${SERVER}/MOPITT/MOP02.003/$k.$i.$j/*.hdf

    done
  done
done
```

Für die Version 4 ist MOP02.003 durch MOP02.004 zu ersetzen. Der Server wechselt ab und an, das lässt sich dann auf der MOPITT-Seite:

http://eosweb.larc.nasa.gov/PRODOCS/mopitt/table_mopitt.html
nachprüfen.

Es kommt auch vor, dass vereinzelt Dateien nachgeschoben werden, es lohnt sich also, routinemäßig das Skript laufen zu lassen.

Die Daten beginnen am 1.1.2001 und sind im Schnitt 14 Tage im Hintertreffen.

Die Benennung der Dateien folgt diesem Schema:

MOP(Prozesslevel)-yyyymmdd-L(Prozesslevel)V(Retrievalversion).(validiert|provisorisch).hdf

z. B. MOP02-20080425-L2V5.93.2.val.hdf

2 Konvertierung von hdf in ncdf mit mopittv4tonc.pro

Mit der IDL-Routine `mopittv4tonc.pro` werden die hdf-Dateien in das ncdf-Format konvertiert, es werden ζ/θ und die Mittagszeit (`noontime`) sowie die Mittagspositionen, nachdem sie mit Trajektorienrechnung bestimmt wurden, hinzugefügt.

Diese Routine ruft folgende Programme/Routinen auf, die nicht der `icglib` oder der `CLaMS`-suite entstammen:

- `get_dates.pro`
- `noontime.pro`
- `create_empty_mopitt_struct.pro`
- `add_zeta.pro` (ruft `clams/clams-tools/pos_add` auf)
- `add_noon_pos_mopitt.pro`
 - `calc_traj.pro` (ruft `clams/traj/traj` auf)
- `check_noonpos.pro`

Die Benennung der Dateien folgt diesem Schema:

`mopitt_L(Prozesslevel)_(Mittagsposition hinzugefügt)_(ECT für θ |ECZ für ζ)_yyymmdd.nc`

z. B. `mopitt_L2_noonpos_ECZ_20010101.nc`

Für die Version 4 lautet die Dateibezeichnung z. B. `mopittv4_L2_noonpos_ECZ_20010101.nc`

Erläuterungen

- in `mopittv4tonc.pro` folgende Variablen anpassen:
 - `dir_in`
 - `dir_out`
 - `startdate`
 - `enddate`
 - `dir_wind`
- `add_zeta`: Aufruf von `pos_add` anpassen
- in `add_noon_pos_mopitt.pro`:
 - Variable `input_dir` anpassen
 - Variable `prefix_wind` anpassen
 - evtl. Übergabeparameter für `calc_traj` ändern
- in `calc_traj.pro`: Aufruf von `traj` anpassen

Aufruf mit Hilfe eines Shell-Skripts

Um IDL-Abstürze in *mopittv4tonc* zu vermeiden, kann das Shell-Skript *run_mopittv4tonc.ksh* genutzt werden. Dieses bearbeitet alle Datensätze in einem angegebenen Zeitraum und ruft das IDL-Programm für jeden Tag erneut auf. Sollte das IDL-Programm abstürzen, wird mit dem nächsten Tag fortgefahren. Das IDL-Programm *mopittv4tonc* wurde entsprechend angepasst und liest das aktuelle Datum von einer im Skript gesetzten Umgebungsvariablen.

- Im Skript *run_mopittv4tonc.ksh*: Start- und Endzeit eintragen
- In IDL: sav-File erstellen (s. *make_sav_file.mopittv4tonc*)
- Aufruf des Shell-Skripts

ACHTUNG: Es tritt ein Fehler beim Aufruf von *write_ncdf* auf. Daher wurde eine lokale Version von *write_ncdf* abgelegt, in der die Ausgabe von Attributen auskommentiert wurde, um den Fehler zu umgehen.

3 Zusammenfassen von jeweils 5 Tagen

`mult_mopitt_synpos.pro`

Mit der IDL-Routine `mult_mopitt_synpos.pro` werden jeweils 5 Tage zusammengefasst und die Positionen mittels Trajektorienrechnung auf einen synoptischen Zeitpunkt gebracht (der jeweils 3. Tag, 12 Uhr).

Diese Routine ruft folgende Programme/Routinen auf, die nicht der `icglib` oder der `CLaMS-suite` entstammen:

- `add_syn_pos_mopitt.pro`
 - `calc_traj.pro`

Die Benennung der Dateien folgt diesem Schema:

`mopitt_L(Prozesslevel)_(ECT für θ |ECZ für ζ)_(synoptische Position hinzugefügt)_yyyymmdd.nc`

z. B. `mopitt_L2_ECZ_synpos_20010103.nc`

Für die Version 4 lautet die Dateibezeichnung z. B. `mopittv4_L2_ECZ_synpos_20010103.nc`

Erläuterungen

- in `mult_mopitt_synpos.pro` folgende Variablen anpassen:
 - `dir_in`
 - `dir_syn`
 - `date_first`:
synoptischer Zeitpunkt, auf den zusammengefasst wird, z. B. für 01.-05.01.:
03.01. angeben (läßt sich mit Hilfe der Routine `calc_begin_of_synpos` ermitteln)
 - `date_last`
 - `prefix_noon`
 - `prefix`
- in `add_synpos_mopitt.pro` folgende Variablen anpassen:
 - `output_dir`
 - `input_dir`
 - `prefix_wind`
- in `calc_traj.pro`: Aufruf von `traj` anpassen

Aufruf mit Hilfe eines Shell-Skripts

Um IDL-Abstürze in `mult_mopitt_synpos` zu vermeiden, kann das Shell-Skript `run_mult_mopitt_synpos.ksh` genutzt werden. Dieses bearbeitet alle Datensätze in einem angegebenen Zeitraum und ruft das IDL-Programm für jeden fünften Tag erneut auf. Sollte das IDL-Programm abstürzen, wird mit dem nächsten Datum fortgefahren. Das IDL-Programm `mult_mopitt_synpos` wurde entsprechend angepasst und liest das aktuelle Datum von einer im Skript gesetzten Umgebungsvariablen.

- Im Skript *run_mult_mopitt_synpos.ksh*: Start- und Endzeit eintragen
- In IDL: sav-File erstellen (s. *make_sav_file.mult_mopitt_synpos*)
- Aufruf des Shell-Skripts

ACHTUNG: Es tritt ein Fehler beim Aufruf von *write_ncdf* auf. Daher wurde eine lokale Version von *write_ncdf* abgelegt, in der die Ausgabe von Attributen auskommentiert wurde, um den Fehler zu umgehen.

4 Gridden der Daten mit `mult_synpos_gridding_mopitt.pro`

Die dem vorangegangenen Schritt entstammenden Daten werden mit der IDL-Routine `mult_synpos_gridding_mopitt.pro` auf ein 2×6 Breiten-Längen-Gitter gebracht und gewichtet gemittelt.

Diese Routine ruft folgende Programme/Routinen auf, die nicht der `icglib` oder der `CLaMS-suite` entstammen:

- `get_cbar.pro`
- `fill_missing_values.pro`

Die Benennung der Dateien folgt diesem Schema:

`grid_(Spezies)_mopitt_(ζ hinzugefügt)_yymmddhh.nc`

z. B. `grid_CO_mopitt_zeta_08022112.nc`

Die Routinen für Version 3 und 4 unterscheiden sich im Algorithmus, jedoch nicht im Namen! Für die Version 4 lautet die Dateibezeichnung z. B. `mopittv4_L2_ECZ_synpos_20010103.nc`

Erläuterungen

- in `mult_synpos_gridding_mopitt.pro` folgende Variablen anpassen:
 - `jssyn`
 - `jsmin`
 - `jsmax`
 - `jsend`
 - `pvfile`
 - `filenames`
 - `out_file`

5 Erzeugen von ζ -Flächen mit `grid2zeta.pro`

Die IDL-Routine `grid2zeta.pro` interpoliert die CO-Werte auf ζ -Flächen.

Diese Routine ruft folgende Programme/Routinen auf, die nicht der `icglib` oder der `CLaMS-suite` entstammen:

- `fill_missing_values.pro`

Die Benennung der Dateien folgt diesem Schema:

`grid_(Spezies)_mopitt_(ζ hinzugefügt)_(interpoliert)_yymmddhh.nc`

z. B. `grid_CO_mopitt_zeta_interpol_08022112.nc`

Erläuterungen

- in `grid2zeta.pro` folgende Variablen anpassen:
 - `start_js`
 - `end_js`
 - `file_press`
 - `file_zeta`
- Zur Erstellung einer Randbedingungsdatei auf Druckniveaus entfällt Schritt 5 !

6 Erstellung der Randbedingungsdatei mit `make_mbound_interpol.pro`

In diesem Schritt wird aus den zuvor erzeugten Dateien der gewünschte Höhenlevel extrahiert und in eine Datei geschrieben, die von `bmix` aus der CLaMS-suite verwendet werden kann.

Diese Routine ruft folgende Programme/Routinen auf, die nicht der `icglib` oder der CLaMS-suite entstammen:

- `create_mbound_struct.pro`

Erläuterungen

- in `make_mbound_interpol.pro` folgende Variablen anpassen:
 - `begindate`
 - `emddate`
 - `all_files`
 - `file_out`
 - `file_out_1_5`

Zur Erstellung einer Randbedingungsdatei auf Druckniveaus muss statt dessen das Programm `make_mbound_press.pro` aufgerufen werden.

Erläuterungen

- in `make_mbound_press.pro` folgende Variablen anpassen:
 - `levs`
 - `begindate`
 - `emddate`
 - `all_files`
 - `file_out`
 - `file_out_1_5`

7 Vorhandene Datensätze

Version 3

Daten in Version 3 nur bis zum 27.07.2009 !

Version 4

Daten in Version 4 nur bis zum 31.12.2012 !

1. Originaldaten in HDF-format:
nicht gespeichert (können schnell wiedergeholt werden!)
2. Nach Netcdf konvertiert und NOONPOS hinzugefügt:
 - /usr/nfs/sat_data_icg1000/mopitt/noonpos
20.02.2008 - 31.12.2008
 - /usr/nfs/clams_icg1000/icg112/mopitt/noonpos
30.12.2008 - 31.10.2012
 - Dateinamen:
mopittv4_L2_noonpos_ECZ_yyyymmdd.nc (bis 2009)
mopittv4_L2_noonpos_ecmwf_yyyymmdd.nc (ab 2010)
3. Jeweils 5 Tage zusammengefaßt und auf einen synoptischen Zeitpunkt gebracht:
 - /usr/nfs/clams_icg1000/icg112/mopitt/synpos
25.09.2001
17.02.2002
16.07.2006 - 31.10.2012
 - Dateinamen:
mopittv4_L2_ECZ_synpos_yyyymmdd.nc (bis 2009)
mopittv4_L2_ecmwf_synpos_yyyymmdd.nc (ab 2010)
4. Auf 2x6 Breiten-Längen-Gitter:
 - /usr/nfs/clams_icg1000/icg112/mopitt/grids
2001 - Oktober 2012
 - Dateinamen: grid_CO_mopittv4_zeta_yymmddhh.nc
5. Schritt 5 für Boundfiles auf Druckniveaus nicht nötig
6. Boundfiles (auf Druckniveaus):
 - /usr/nfs/clams_icg1000/icg112/mopitt/boundfiles

8 Laufzeiten

Schritt 2	5 min pro Datensatz/Tag	-> 30 Std. pro Jahr
Schritt 3	1 Std. pro Datensatz (für 5 Tage)	-> 75 Std. pro Jahr
Schritt 4	25 min pro Datensatz (für 5 Tage)	-> 30 Std. pro Jahr