

a. Shell and CDO

This is a tutorial of how to select certain parts of raw ICON model output, do some simple computation on it, and save/merge the result into one netcdf-file that allows a more convenient data handling. Individual steps that can be directly executed from the shell prompt are listed and explained below.

As a first step, it might be useful to inspect the different model files of the simulation that you are interested in by using `ncdump`. This lists information about the specific file type. For example for one output file of the `dpp0016` simulation, whereby `dpp0016` is a global coupled ICON simulation run with a grid spacing of 5km:

```
ncdump -h /work/mh0287/k203123/GIT/icon-aes-  
dyw_albW/experiments/dpp0016/dpp0016_atm_2d_ml_20200201T000000Z.nc
```

The output will look like:

```
netcdf dpp0016_atm_2d_ml_20200201T000000Z {  
dimensions:  
  ncells = 20971520 ;  
  vertices = 3 ;  
  time = UNLIMITED ; // (4 currently)  
variables:  
  double time(time) ;  
    time:standard_name = "time" ;  
    time:units = "hours since 2020-2-1 00:00:00" ;  
    time:calendar = "proleptic_gregorian" ;  
    time:axis = "T" ;  
  float ps(time, ncells) ;  
    ps:standard_name = "surface_air_pressure" ;  
    ps:long_name = "surface pressure" ;  
    ps:units = "Pa" ;  
    ps:param = "0.3.0" ;  
    ps:CDI_grid_type = "unstructured" ;  
    ps:number_of_grid_in_reference = 1 ;  
  float psl(time, ncells) ;  
    psl:standard_name = "mean sea level pressure" ;  
    psl:long_name = "mean sea level pressure" ;  
    psl:units = "Pa" ;  
    psl:param = "1.3.0" ;  
    psl:CDI_grid_type = "unstructured" ;  
    psl:number_of_grid_in_reference = 1 ;  
  
  ...  
  ...  
  
  float ts(time, ncells) ;  
    ts:standard_name = "surface_temperature" ;  
    ts:long_name = "surface temperature" ;  
    ts:units = "K" ;
```

```
ts:param = "0.0.0" ;
ts:CDI_grid_type = "unstructured" ;
ts:number_of_grid_in_reference = 1 ;

...
...

float tas(time, ncells) ;
tas:standard_name = "tas" ;
tas:long_name = "temperature in 2m" ;
tas:units = "K" ;
tas:param = "0.0.0" ;
tas:CDI_grid_type = "unstructured" ;
tas:number_of_grid_in_reference = 1 ;

// global attributes:
:CDI = "Climate Data Interface version 1.8.3rc
(http://mpimet.mpg.de/cdi)" ;
:number_of_grid_used = 15 ;
:grid_file_uri =
"http://icon-downloads.mpimet.mpg.de/grids/public/mpim/0015/icon_grid_0015_R
02B09_G.nc" ;
:uuidOfHGrid = "0f1e7d66-637e-11e8-913b-51232bb4d8f9" ;
:history = "/work/mh0287/k203123/GIT/icon-aes-dyw_albW/bin/icon at
20200513 234552" ;
:source = "git@gitlab.dkrz.de:icon/icon-
aes.git@b582fb87edbd30b10a36223d10fbd0c20f31dee6" ;
:institution = "Max Planck Institute for Meteorology" ;
:Conventions = "CF-1.6" ;
:title = "ICON simulation" ;
:references = "see MPIM/DWD publications" ;
:comment = "Sapphire Dyamond (k203123) on m21335 (Linux
2.6.32-754.14.2.el6.x86_64 x86_64)" ;
}
```

Next, as an example, say you are interested in reading surface temperature data (ts) from dpp0016 for February and store it as daily mean values in one single netcdf file. In the example shown below, the output will be stored in the \$SCRATCH-directory of the user in order to avoid accidentally overloading the users \$WORK-directory:

```
cdo -daymean -cat -apply,selname,ts [ /work/mh0287/k203123/GIT/icon-aes-
dyw_albW/experiments/dpp0016/dpp0016_atm_2d_ml_202002*.nc ]
$SCRATCH/dpp0016_atm_2d_ml_TS.nc
```

From:

<https://wiki.mpimet.mpg.de/> - **MPI Wiki**

Permanent link:

https://wiki.mpimet.mpg.de/doku.php?id=analysis:postprocessing_icon:1._open_read:cdo:start

Last update: **2020/09/24 20:54**

