

The folder contains:

README.md
doConRemapCORDEX_v000.sh
grid_bounds_calc.py
source.grid_EUR44_ALADIN52
source.grid_EUR44_ALADIN53
source.grid_EUR44_ALARO-0
source.grid_NAM22_NCAR-RegCM4
source.grid_NAM22_NCAR-WRF
source.grid_NAM22_NCAR-WRFH
source.grid_NAM44_NCAR-RegCM4
source.grid_NAM44_NCAR-WRF

doConRemapCORDEX_v000.sh: The main script for interpolation

Naming format of the CORDEX input files:

<CORDEX-
domain>_<forcing_GCM>_<ensemble_member>_<experiment>_<RCM>_<version>_<variable>_
<frequency>_<period>.nc4"

ESGF CORDEX naming format:

<variable>_<domain>_<forcing_GCM>_<experiment>_<ensemble_member>_<RCM>_<version>_
<frequency><period>.nc"

The script has 3 loops:

- (1) through variables (defined by a user)
- (2) through RCMs (determined by the script itself from the input folder defined by a user)
- (3) through each file with the corresponding RCM (determined by the script itself from the input folder defined by a user)

The script has to main parts:

- (1) CHANGE AND ADAPT TO YOUR REQUIREMENTS:

Here a user gives all the info necessary for the script to run, such as the domain, domain boundaries, paths etc.

- (2) NO CHANGES NECESSARY:

This part of the script is a core part and no changes should be introduced here. It is divided in 4 subparts:

- (a) Preparations for the remapping
- (b) Creating the source.grid file
- (c) Conservation remapping
- (d) Finish: moving and renaming files, loop endings

(a) Preparations for the remapping

For each variable the script creates files necessary for the interpolation:

1. *refmask.nc4* - shrunk destination masked fitting the source grid
2. *destination.grid* - information on the destination grid (cdo griddes)
3. *filelist.txt* - list of all files in the input folder
4. *models* - list of all RCM models in the input folder (set to read the 5th place from the filename of the CORDEX input file)

The script checks if a mask for the source RCM exists. If source mask does not exist, basic interpolations is done without any dependency on the processed variable.

When the mask exists, the script checks if the variable has a strong land-sea contrast. If yes, then the land-sea correction is done, if not then just a basic interpolation is done.

Variables with the strong land-sea contrast: tas, tasmin, tasmax, huss, sfcWind, sfcWindmax, mfrso, mrros, mrro, mrso, snw, snm, uas, vas, snc, snd, sic, evspsbl, hfss, hfls

(b) Creating the source.grid file

The script first check if a source.grid_<GCM> exists in the folder. If yes, then file will be copied to source.grid. If not, than the script check if the filename that is processed contains necessary info. This is done in a way that it checks a folder where original, not postprocessed files are located. If a file that correspond to the RCM exist, this file will be used for creating the source.grid file using cdo griddes command.

If the folder doer not contain any file with the corresponing grid, the file that is currently postporcessing will be used for creating sourc.grid file.

(c) Conservation remapping

In this part the script recognizes if the land-sea contrast correction will be done or not, by setting a MS_STYLE variable to true of false.

Land-sea contrast is done in a way to do the interpolation separately over the land and over the water surfaces. The gaps between the land and sea are recognized, and then filled with the third-basic interpolation that is previously done.

The interpolation is done using the cdo command for remapping

(d) Finish: moving and renaming files, loop endings

All the 3 loops are ended, some files between loops that are needed to be deleted are deleted, the output file is relocated from the working directory to the output folder, and the final file will be renamed. The name of the output file will be the same as the input file, only the domain name will have added "i" (e.g. EUR44 --> EUR44i)

NOTE:

ALARO-0, ALADIN52, ALADIN53, NCAR-WRF and NCAR-RegCM4 RCMs use the Lambert Conformal Conilcal (LCC) projection. For Interpolation, the python program **grid_bounds_calc.py** is run before the main script *doConRemapCORDEX_v000.sh*. The python program is obtained from Meteo-France, courtesy of CAILLAUD Cécile (cecile.caillaud@meteo.fr) and Samuel Somot (samuel.somot@meteo.fr). The script calculates grid corners for each grid cell, necessary for the conservative interpolation.

The steps for RSMs with non-rotated projection, e.g. with LCC projection:

Change 2 lines referring to a reference file for a specific RCM --> the path and the filename.

```
$ python3 grid_bouds_calc.py ( → file_out created in the working directory)
```

```
$ cdo griddes file_out > source.grid_<domain_name>_<name of the source model>
```

(**source.grid_<domain_name>_<name of the source model>** files in the folder are source.grid files for all recognized 0.44 and 0.22 CORDEX RCMs with non-rotated projections)

After these 2 steps, run the main interpolation script *doConRemapCORDEX_v000.sh*:

```
$ ./ doConRemapCORDEX_v000.sh
```

Contact person: milovacj@unican.es