PyCordexer

A RegCM output format converter according to CORDEX archive specifications

December 2014

The Abdus Salam
International Centre
for Theoretical Physics

The *PyCordexer* scripts have been developed to ease the RegCM Model User in converting variables in RegCM model output files into the netCDF format respecting the CORDEX experiment conventions.

It has been developed to satisfy the requirements in the 05/03/2014 release of the CORDEX Specification Document available from the DMI site at:

http://cordex.dmi.dk/joomla

The *PyCordexer* script have been developed by ICTP staff to post-process the CREMA experiment output dataset to be contributed to the CORDEX archive, and have been adapted afterwards to post-process the latest RegCM model output formats.

Any output file of the RegCM model from version 4.3 onward should be compatible with the *PyCordexer* scripts.

1 Requirements

To run the python script, you need a working Python2 interpreter (2.7 is suggested) and netCDF4 Python library:

https://unidata.github.io/netcdf4-python

The package requirement list for the library can be found at in the above site.

2 Installation

Part of the interpolation and derived variable calculation has been taken from RegCM Model source code, and you need to compile the fortran source codes in the *PyCordexer* directory using the f2py program of the Python numpy package, which is a requirement of the netcdf4-python module. Just type make in the *PyCordexer* directory to compile the fortran code to be used by python

```
cd pycordexer
make
```

Adapt the Makefile in the directory if the numpy python has been compiled with a different compiler.

The scripts can be run from the *PyCordexer* directory afterwards.

3 PyCordexer Scripts

The *PyCordexer* is split into two separate scripts:

1. **cordex.py** This script extracts variables from RegCM output files and creates a new file in the CORDEX netCDF format.

2. **means.py** This script computes day and monthly averages of CORDEX files at higher temporal resolution saving them in CORDEX netCDF format files.

4 The cordex.py script

The **cordex.py** script reads a RegCM model output file and extracts the variable creating a CORDEX file out of the details specified from the command line arguments.

The syntax to invoke the program is:

```
cordex.py datafile variable [mail domain model experiment ensemble notes [corrflag]]
```

The *datafile* is a RegCM file which has the variable specified by *variable* inside.

4.1 List of Implemented Variables

The variable which can be extracted from the RegCM file are:

variable	RegCM file	Description
tas	SRF,STS	Near-Surface Air Temperature
pr	SRF,STS	Precipitation
prc	SRF	Convective Precipitation
huss	SRF	Near-Surface Specific Humidity
hurs	SRF	Near-Surface Relative Humidity
evspsbl	SRF	Evaporation
mrros	SRF	Surface Runoff
ps	SRF	Surface Air Pressure
psl	ATM	Sea Level Pressure
tasmax	STS	Daily Maximum Near-Surface Air Temperature
tasmin	STS	Daily Minimum Near-Surface Air Temperature
sfcWindmax	STS	Daily Maximum Near-Surface Wind Speed
mrro	SRF	Total Runoff
sfcWind	SRF	Near-Surface Wind Speed
ua850	ATM,ATMp	Eastward Wind (at 850 hPa)
va850	ATM,ATMp	Northward Wind (at 850 hPa)
ta850	ATM,ATMp	Air Temperature (at 850 hPa)
hus850	ATM,ATMp	Specific Humidity (at 850 hPa)
ua500	ATM,ATMp	Eastward Wind (at 500 hPa)
va500	ATM,ATMp	Northward Wind (at 500 hPa)
ta500	ATM,ATMp	Air Temperature (at 500 hPa)
zg500	ATM,ATMp	Geopotential Height (at 500 hPa)
ua200	ATM,ATMp	Eastward Wind (at 200 hPa)
va200	ATM,ATMp	Northward Wind (at 200 hPa)
ta200	ATM,ATMp	Air Temperature (at 200 hPa)
zg200	ATM,ATMp	Geopotential Height (at 200 hPa)

4.2 Arguments

The first two arguments are required, while the other have basic defaults which do not comply to CORDEX standard but allow the script to be used as a generic post-processing tool for RegCM

Argument	Default	Meaning
mail	esp@ictp.it	E-Mail of the Person responsible of this run
domain	NONE	CORDEX domain (EUR-44,AFR-44,etc)
model	NONE	CMIP5 driving model (MOHC-HadGEM2-ES,etc)
experiment	none	CMIP5 experiment (historical,rcp85,etc)
ensemble	NN	CMIP5 model experiment (r1i1p1,r2i1p1,etc)
notes	none	Eventual notes to be added in the file
corrflag	1	Apply a correction time to dates.

4.3 Usage

Example usage is the following. Suppose you have in a directory RegCM model output file, and want to extract in CORDEX format the variable tas from SRF file for the AFR-44 domain run on ECMWF ERA15 dataset:

```
cordex.py /data/output/path/Africa_SRF.2002030100.nc tas me@here
AFR-44 ECMWF-ERAINT evaluation r1i1p1 'Some text'
```

This will create in the current directory the file:

```
tas_AFR-44_ECMWF-ERAINT_evaluation_r1i1p1_ICTP-RegCM4-
3_v4_3hr_200203010300-200204010000.nc
```

with all the mandatory attributes and naming conventions required by CORDEX experiment convention document.

If just the name of the file and the variable name are specified, the output file name will be:

```
tas_NONE_NONE_none_NN_ICTP-RegCM4-3_v4_3hr_200203010300-200204010000.nc
```

5 The means.py script

The **means.py** script reads a CORDEX file created by the **cordex.py** script and computes day or monthly average of the variable in the file as instructed by command line argument, saving the result in a CORDEX conforming netCDF file.

The syntax to invoke the program is:

```
means.py datafile [mon/day]
```

The *datafile* is a file produced by cordex.py, the secund argument if not specified defaults to mon, i.e. creates a monthly average file.

5.1 Usage

Example usage is the following. Suppose you have in a directory the file

```
tas_NONE_NONE_none_NN_ICTP-RegCM4-3_v4_3hr_200203010300-200204010000.nc
```

and want to compute daily averages in a CORDEX conforming netCDF file.

```
means.py tas_NONE_NONE_none_NN_ICTP-RegCM4-3_v4_3hr_200203010300-200204010000.nc day
```

This will create in the current directory the file:

```
tas_NONE_NONE_none_NN_ICTP-RegCM4-3_v4_day_2002030112-2002033112.nc
```

which contains daily mean values with all the mandatory attributes and naming conventions required by CORDEX experiment convention document.

If you want to compute monthly average, you can issue:

```
means.py tas_NONE_NONE_none_NN_ICTP-RegCM4-3_v4_day_2002030112-2002033112.nc mon
```

and obtain:

```
tas_NONE_NONE_none_NN_ICTP-RegCM4-3_v4_mon_20020301-20020331.nc
```

6 Contacts

The *pycordexer* scripts are now kept in the *Scripts/Tools* directory of the RegCM model code package, and are maintained by ICTP as part of the RegCM codebase. Please address any problem/suggestion to the RegCNET mailing list:

https://lists.ictp.it/mailman/listinfo.cgi/regcnet