Poster
Y15

Vienna, Austria, April 2015
Wednesday, 15 Apr 2013
author in attendance: 17:30–19:00

Using the Python language and the CMOR2 library to create PMIPn-CMIPn compliant model output

Jean-Yves Peterschmitt[1], Charles Doutriaux

[1] Laboratoire des Sciences du Climat et de l'Environnement (LSCE), Gif-sur-Yvette, France, *jean-yves.peterschmitt@lsce.ipsl.fr*

[2] PCMDI, Lawrence Livermore National Laboratory, Livermore, CA, USA, *doutriaux1@llnl.gov*

Abstract EGU2015-4484

The Paleoclimate Modelling Intercomparison Project (PMIP) is a long standing initiative that has provided an efficient mechanism for coordinating paleoclimate modelling activities that provide valuable information on the mechanisms of climate change, the identification of key feedbacks operating in the climate system and, through model evaluation, the capability of climate models to reproduce climates different from today. The third phase of PMIP (aka PMIP 3) started in 2009 (the fourth phase is about to start) and followed the requirements specified by CMIP5 (Coupled Model Intercomparison Project).

Generating data files following strict Model Intercomparison Projects (MIPs) standards (NetCDF format, file and variable names, file structure, metadata information, directory hierarchy, etc. . .) has been a key to the success of many recent Model Intercomparison Projects. It is unfortunately not always easy to convert proprietary model output format to the required standards, and this has prevented some smaller modelling groups from sharing their data.

We will present how the Python version of the CMOR2 (Climate Model Output Rewriter) library bundled with the UV-CDAT Python distribution (Ultrascale Visualization Climate Data Analysis Tools) can be used to easily convert raw model output to the appropriate MIP shareable format.

Model Intercomparison Projects (MIPs) … and Conventions

Why are conventions useful?

- Local data in custom format → NOT SO GOOD — only a small group can use the data ☹
- Data following a known convention → BETTER — sharing data is easier
- Distributed standard data → PERFECT — optimal sharing! ☺
  - More science, more (intercomparison) papers
  - More bugs found (and corrected!)
- But ***conventions* means more (data processing) work** …

The CMIP5 convention

- It was a key to the IPCC AR5 success!
- 3 documents for the convention:
  - The format (file names and structure, required metadata, …)
    - *CMIP5 Model Output Requirements: File Contents and Format, Data Structure and Metadata*
    - *CMIP5 Data Reference Syntax (DRS) and Controlled Vocabularies*
  - The data (list of all the requested variables)
    - *standard_output.xls*
- Convention summary
  - Files are in netCDF format
  - The metadata (variable names and attributes) follow the CF (*Climate and Forecast*) convention
  - There is 1 variable per file
    - huge variables can be split into several files, along the time axis
- Details and documents are available on the CMIP5 web site
  - http://cmip-pcmdi.llnl.gov/cmip5/output_req.html?submenuheader=2#req_format

What next?

- CMIP6 and PMIP will use an updated version of the convention

Paleoclimate Modelling
PMIP3
Intercomparison Project

A summary of the DRS (*Data Reference Syntax*)

- **Directory structure**
  <activity>/<product>/<institute>/<model>/<experiment>/<frequency>/
  <modeling realm>/<variable name>/<ensemble member>/<filename>.nc
- **File names**
  <variable name>_<MIP table>_<model>_<experiment>_
  <ensemble member>[_<temporal subset>].nc
  with <temporal subset> = 'yyyy[mm[dd[hh][mm]]][-clim]'
- The fields of the DRS can only have **predefined values** (aka *Controlled Vocabulary*)
  - **activity**: *PMIP3* or *CMIP5*
  - **product**: *output* (PMIP3) or *output1* or *output2* (CMIP5)
  - **institute**: one of the expected contributing groups
  - **model**: one of the expected models
  - **experiment**: standardized short name of one of the PMIP3/CMIP5 experiments
    eg : *piControl*, *midHolocene*, *lgm*, *past1000*, *1pctCO2*, …
  - **frequency**: *fx*, *yr*, *monClim* (climatological monthly mean), *mon*, *day*, *6hr*, *3hr* or *subhr*
  - **modeling realm**: *atmos*, *ocean*, *land*, landIce, *sealce*, *aerosol*, *atmosChem*, *ocnBgchem*
  - **variable name**: the standardized short name of a variable (eg *tas*, *ua*, *va*, *landCoverFrac*, …)
  - **MIP table**: the CMOR2 table where the variable is defined (eg *fx*, *Amon*, *Aclim*, …)
  - **ensemble member**: usually *r0i0p0* for fx fields and *r1i1p1* for the rest. Other values for ensemble runs
  - **version number**: the version of the dataset where the variable was found (usually a *yyyymmdd* date string)
- Example
  PMIP3/output/IPSL/IPSL-CM5A-LR/lgm/
  monClim/atmos/Aclim/r1i1p1/v20120418/
  tas/tas_Aclim_IPSL-CM5A-LR_lgm_r1i1p1_260101-280012-clim.nc

The CMIP5_Amon table…

```
axis_entry:                 longitude
|----------------------------------
! Axis attributes:
|----------------------------------
standard_name:  longitude
units:          degrees_east
axis:           X          ! X, Y, Z, T (default: undeclared)
long_name:      longitude
|----------------------------------
! Additional axis information:
|----------------------------------
out_name:                   lon
valid_min:                   0.0
valid_max:                   360.0
stored_direction:           increasing
type:                       double
must_have_bounds:           yes

variable_entry:             ts

modeling_realm:             atmos
|----------------------------------
! Variable attributes:
|----------------------------------
standard_name:              surface_temperature
units:                      K
cell_methods:               time: mean
cell_measures:              area: areacella
long_name:                  Surface Temperature
comment:          ""skin"" temperature (i.e., SST for open ocean)
|----------------------------------
! Additional variable information:
|----------------------------------
dimensions:                 longitude latitude time
out_name:                   ts
type:                       real
valid_min:                  176.8
valid_max:                  339.6
ok_min_mean_abs:            262.8
ok_max_mean_abs:            293.3
```

```python
#!/usr/bin/env python

import numpy, cdms2, cmor

# Generate a dummy data matrix with '32 degF' everywhere except 2
# latitudinal bands around the equator
raw_data = numpy.ma.ones((1, 10, 18), numpy.float32) * 32
raw_data[:, 4:6, :] = numpy.ma.masked

# Generate axes for the data matrix
lat_axis = cdms2.createUniformLatitudeAxis(90, 10, -20)
lon_axis = cdms2.createUniformLongitudeAxis(-180, 18, 20)

cmor.setup(inpath='/home/users/jypeter/PMIP3_CMIP5/CMOR/cmip5-cmor-tables/Tables/',
           netcdf_file_action=cmor.CMOR_REPLACE)

cmor.dataset(experiment_id='lgm',
             institution='Institut Pierre Simon Laplace',
             institute_id='IPSL',
             model_id='ICE-MAKER-1',
             source='ICE-MAKER-1 atmosphere: JYP-007; ice: AMORINO-1',
             calendar='360_day',
             outpath='.',
             realization=1,
             forcing='N/A',
             parent_experiment_id='N/A', parent_experiment_rip='N/A',
             branch_time=0.,
             contact='JYP@LSCE',
             comment='This is some really dummy data',
             references='100 ways to freeze your ocean, The Ice Cream Journal')

cmor.load_table('CMIP5_Amon')

time_id = cmor.axis(table_entry= 'time',
                    units= 'days since 2000-01-01 00:00:00')

lat_id = cmor.axis(table_entry= 'latitude',
                   units= 'degrees_north',
                   coord_vals=lat_axis[:],
                   cell_bounds=lat_axis.getBounds())
lon_id = cmor.axis(table_entry= 'longitude',
                   units= 'degrees_east',
                   coord_vals=lon_axis[:],
                   cell_bounds=lon_axis.getBounds())

data_id = cmor.variable(table_entry='ts',
                        units='degF',
                        axis_ids=[time_id, lat_id, lon_id],
                        missing_value=raw_data.fill_value,
                        original_name='temp',
                        comment='A dummy variable from a dummy dataset')

# Write 3 time steps to the file (January to March). Add 18 degF to
# each time step (which should be converted to 10 K in the output
# file)
for month in range(3):
    cmor.write(var_id=data_id,
               data=raw_data + month * 18,
               time_vals=numpy.array([15. + 30 * month]),
               time_bnds=numpy.array([0. + 30 * month, 30. + 30 * month])
)

file_path = cmor.close(data_id, file_name=True)

print '\nThe dummy data was successfully saved to', file_path
```

CMOR, python and CDAT to the rescue!

What is the Climate Model Output Rewriter?

- "*CMOR was not designed to serve as an all-purpose writer of CF-compliant netCDF files, but simply to reduce the effort required to prepare and manage MIP model output*"
- CMOR is a library that can be called from Fortran, C **and python**
- CMOR uses MIP-specific ***CMOR tables*** to define the requested variables and meta-data
- CMOR will automatically re-arrange the axes and change the units to follow the CMOR tables requirements
- CMOR will do some basic quality control of the converted variables (variable range check, …)

Why use Python (rather than shell script and Fortran)?

- Python is an interpreted language and makes it easy to test functions interactively and prototype data processing frameworks
- Any standard Python is very good with:
  - string handling (generating file names!),
  - making efficient system calls (finding the files!),
  - reading command line parameters and configuration files (what do I want to do with which files?),
  - making loops (processing all the files),
  - calling external programs (if you really have not found a way to do it in Python), …

Why use UV-CDAT for preparing MIP data?

- It is a Python distribution specifically targeted at climate scientists
  - UV-CDAT ➔ Ultra Scale Climate Data Analysis Tools
- UV-CDAT provides the *python modules* we need for preparing MIP data files
  - cmor: the library and the Python wrapper
  - cdms2/cdtime: powerful netCDF I/O (for climate scientists) and time axes handling
  - cdutil/genutil: utilities for easy climate data massaging (eg easy accurate spatial and temporal averaging)
  - vcs: easy climate data plotting
  - More bundled bells and whistles (regridding, plotting, …)

```
netcdf ts_Amon_ICE-MAKER-1_lgm_r1i1p1_200001-200003 {
dimensions:
        time = UNLIMITED ; // (3 currently)
        lat = 10 ;
        lon = 18 ;
        bnds = 2 ;
variables:
        double time(time) ;
                time:bounds = "time_bnds" ;
                time:units = "days since 2000-01-01 00:00:00" ;
                time:calendar = "360_day" ;
                time:axis = "T" ;
                time:long_name = "time" ;
                time:standard_name = "time" ;
        double time_bnds(time, bnds) ;
        double lat(lat) ;
                lat:bounds = "lat_bnds" ;
                lat:units = "degrees_north" ;
                lat:axis = "Y" ;
                lat:long_name = "latitude" ;
                lat:standard_name = "latitude" ;
        double lat_bnds(lat, bnds) ;
        double lon(lon) ;
                lon:bounds = "lon_bnds" ;
                lon:units = "degrees_east" ;
                lon:axis = "X" ;
                lon:long_name = "longitude" ;
                lon:standard_name = "longitude" ;
        double lon_bnds(lon, bnds) ;
        float ts(time, lat, lon) ;
                ts:standard_name = "surface_temperature" ;
                ts:long_name = "Surface Temperature" ;
                ts:comment = "A dummy variable from a dummy dataset, CMIP5_table_comment: \"\"skin\"\" temperature (i.e., SST for open ocean)" ;
                ts:units = "K" ;
                ts:original_name = "temp" ;
                ts:original_units = "degF" ;
                ts:history = "2015-04-08T12:20:19Z altered by CMOR: Converted units from \'degF\' to \'K\'. 2015-04-08T12:20:19Z altered by CMOR: Inverted axis: lat." ;
                ts:cell_methods = "time: mean" ;
                ts:cell_measures = "area: areacella" ;
                ts:missing_value = 1.e+20f ;
                ts:_FillValue = 1.e+20f ;
                ts:associated_files = "baseURL: http://cmip-pcmdi.llnl.gov/CMIP5/dataLocation gridspecFile: gridspec_atmos_fx_ICE-MAKER-1_lgm_r0i0p0.nc areacella: areacella_fx_ICE-MAKER-1_lgm_r0i0p0.nc" ;

// global attributes:
                :institution = "Institut Pierre Simon Laplace" ;
                :institute_id = "IPSL" ;
                :experiment_id = "lgm" ;
                :source = "ICE-MAKER-1 atmosphere: JYP-007; ice: AMORINO-1" ;
                :model_id = "ICE-MAKER-1" ;
                :forcing = "N/A" ;
                :parent_experiment_id = "N/A" ;
                :parent_experiment_rip = "N/A" ;
                :branch_time = 0. ;
                :contact = "JYP@LSCE" ;
                :comment = "This is some really dummy data" ;
                :references = "100 ways to freeze your ocean, The Ice Cream Journal" ;
                :initialization_method = 1 ;
                :physics_version = 1 ;
                :tracking_id = "1bf69996-28f5-4654-b21e-1dc7e1109da1" ;
                :product = "output" ;
                :experiment = "last glacial maximum" ;
                :frequency = "mon" ;
                :creation_date = "2015-04-08T12:20:19Z" ;
                :history = "2015-04-08T12:20:19Z CMOR rewrote data to comply with CF standards and CMIP5 requirements." ;
                :Conventions = "CF-1.4" ;
                :project_id = "CMIP5" ;
                :table_id = "Table Amon (26 July 2011) 976b7fd1d9e3be31dddd28f5dc79b7d1" ;
                :title = "ICE-MAKER-1 model output prepared for CMIP5 last glacial maximum" ;
                :parent_experiment = "N/A" ;
                :modeling_realm = "atmos" ;
                :realization = 1 ;
                :cmor_version = "2.9.1" ;
```

```
lon = 0, 20, 40, 60, 80, 100, 120, 140, 160, 180, 200, 220, 240, 260, 280,
    300, 320, 340 ;

lon_bnds =
  -10, 10,
  10, 30,
  30, 50,
  [...]
  310, 330,
  330, 350 ;

ts =
  273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15,
    273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15,
  [...]
  273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15,
    273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15,
  _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
  _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
  273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15,
    273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15,
  [...]
  273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15,
    273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15, 273.15,
    283.15, 283.15, 283.15, 283.15, 283.15, 283.15, 283.15, 283.15, 283.15,
  [...]
  _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
  _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
  283.15, 283.15, 283.15, 283.15, 283.15, 283.15, 283.15, 283.15, 283.15,
  [...]
```

References

- CMIP5 output — http://cmip-pcmdi.llnl.gov/cmip5/output_req.html
- CF convention — http://cfconventions.org/
- CMOR — http://www2-pcmdi.llnl.gov/cmor
- UV-CDAT — http://uvcdat.llnl.gov/
- Python — https://www.python.org/
- PMIP3 — http://pmip3.lsce.ipsl.fr/