



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

Jean-Yves Peterschmitt (1) and Charles Doutriaux (2)

(1) CEA / CNRS, LSCE / IPSL, Gif-sur-Yvette, France (jean-yves.peterschmitt@lsce.ipsl.fr), (2) PCMDI, Lawrence Livermore National Laboratory, Livermore, CA, USA (doutriaux1@llnl.gov)

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.

References:

<http://pmip3.lsce.ipsl.fr/>
http://cmip-pcmdi.llnl.gov/cmip5/output_req.html
<http://www2-pcmdi.llnl.gov/cmor>
<http://uvcdat.llnl.gov/>
<https://www.python.org/>