

1 Overview

In this lecture we will learn to know various helpful tools and techniques that are used in the analysis and processing of gridded climate data that is — for example — provided by climate models. First, we will learn to know the NetCDF file format in a few short practical demonstrations. Second, some analyses will be performed on available gridded climate data.

Topics that this lecture considers are:

- **Network Common Data Form (NetCDF)**: general information on the topic and how to use NetCDF files
- **Climate Data Operators (CDO)**: several examples of data reduction, analysis and transformation of NetCDF files, including operator piping
- **Xarray**: simple python package that had similar uses as CDO

We will use the pre-installed environment made on University of Bremen's jupyter hub for the purpose of this lecture. This is possible by using the cdo package that allows us to use python. But keep in mind that usually used or are built to use on command line interfaces

1.1 A very limited introduction to tools and methods employed in this exercise

Below some information regarding tools and methods, that are relevant for this exercise, is collected in the form of a very general overview. This information collection is intended as a supplement to the lecture. A few complementary information sources are listed in the subsection "Further Reading". This list is obviously far from being complete. There is a vast amount of freely-accessible information available on-line and the interested student is kindly invited to use search engines towards locating and exploring further documentation.

1.1.1 NetCDF

A very precise definition on the characteristic and purpose of NetCDF is given in the NetCDF FAQ, "What is netCDF?" (for a link, see section "Further Reading"):

"NetCDF (network Common Data Form) is a set of interfaces for array-oriented data access and a freely distributed collection of data access libraries for C, Fortran, C++, Java, and other languages. The netCDF libraries support a machine-independent format for representing scientific data. Together, the interfaces, libraries, and format support the creation, access, and sharing of scientific data."

NetCDF is a data-container that has been established as a widely used file-standard in science and engineering. It has been developed for storing array-oriented values in compact and interchangeable files. The most important characteristics of NetCDF files can as well be found in the NetCDF FAQ, “What is netCDF?”. An excerpt, that highlights the advantages of NetCDF with respect to this exercise, is listed here:

- Self-Describing. A NetCDF file includes a description of the data that it contains.
- Portable. A NetCDF file can be accessed by computers that apply different formats of storing integers, characters, and floating-point numbers.
- Scalable. A small subset of a large dataset may be accessed efficiently.
- Appendable. Data may be appended to a properly structured NetCDF file without copying the dataset or redefining its structure.

These characteristics make NetCDF a perfect choice for storing any kind of array-oriented data. The data form that we will work with in this exercise is as well array-oriented - therefore, we will use NetCDF as data container for both input and output of computations.

Since NetCDF is a binary format (in contrast to ASCII-text, which can be examined and edited by means of any common text editor), reading, writing and changing of NetCDF files necessitates the use of dedicated software. Fortunately, such software is freely available and can easily be installed on any UNIX system. The following tools are of particular importance:

- ncview (lightweight but mighty explorer for NetCDF files)
- ncbrowse (a Java-based alternative to ncview for Windows)
- Panoply (a flexible Java-based generator of geographic maps of NetCDF data)
- ncdump (tool for “dumping” the contents of a NetCDF file to human-readable ASCII-text; the complete description and structure of the NetCDF file is preserved)
- ncgen (complementary to ncdump, generates a binary NetCDF file from a NetCDF ASCII-dump)
- ncks (mighty toolbox for modification of NetCDF files)
- cdo (mighty toolbox for analysis and modification of NetCDF files, strong focus on climatological data)

1.1.2 CDO

While NetCDF defines a file format (and supporting programs and routines) that can store climatological data in a practical way, the CDO are a collection of operators that allow analysis and modification of gridded binary climatological data. In climate sciences, the CDO have become a very common software tool due to the vast number of available operators and their flexibility:

- more than 400 designated operators are available
- operator-piping allows the application of complex methods on climatological data in a compact manner
- the CDO are command-line programs; in combination with shell-scripts they can be automated, and enhanced complexity of data processing and analysis may be achieved

CDO operator piping in combination with shell-programming can be demonstrated shortly in the following very short bash-script, where two input files are interpolated to a common resolution, the resulting fields are added, the sum is time-averaged, and the result being stored in a new file. Shell-programming allows for the diagnostic output of additional information to the screen, here the spatial-average of the field resulting from the CDO-operator-chain.

```
1 #!/bin/bash
2
3 #select level 6, interpolate to 1x1 degree, and convert from deg. C to Kelvin
4 cdo timmean -addc,273.15 -remapcon,r360x180 -sellevel,6 input.nc output.nc
5 #note: the rightmost command is executed first!
6
7 #compute global mean
8 spat_avg=$(cdo output -fldmean output.nc)
9
10 #print result to screen
11 echo
12 echo
13 echo "spatial-average-of-global-ocean-surface-temperature-is-${spat_avg}-K."
14
15 #clean up
16 rm output.nc
```

1.1.3 Xarray

While NetCDF defines a file format (and supporting programs and routines) that can store climatological data in a practical way, **xarray** is a Python-based library that allows analysis and manipulation of gridded, labeled multi-dimensional climatological data. In combination with tools such as the *CDO*, which provide command-line operators for climate data processing, xarray enables flexible and interactive data analysis within the Python ecosystem.

In climate sciences, xarray has become a very common software tool due to its intuitive handling of multi-dimensional labeled arrays and its integration with scientific Python libraries:

- supports labeled multi-dimensional arrays with named dimensions such as time, latitude, and longitude
- allows direct selection and manipulation of data using labels instead of numeric indices
- provides high-level operations such as averaging, grouping, resampling, and reduction along dimensions
- integrates directly with NetCDF file formats for reading and writing climatological datasets
- works seamlessly with NumPy, Matplotlib, and Dask for computation, visualization, and large-scale processing

In contrast to command-line tools such as CDO, xarray operates entirely within Python, allowing interactive workflows and reproducible analysis pipelines. Complex processing steps can be expressed in a compact and readable way using chained operations on datasets.

1.2 The Bash, a popular UNIX-Shell

Due to their scripting ability Shells are a very powerful tool of UNIX-systems. This and the fact, that a large number of (free) software tools and programs are available on UNIX systems, makes UNIX-computers the choice for tackling complex scientific problems that involve the analysis and processing of large amounts of data. Particularly the bash-shell is a famous tool for scientists and commonly used during their daily work routine. Yet, giving a comprehensive overview on the use and ability of the bash clearly exceeds the scope of this course. Please refer to a bash-scripting guide referenced below if you would like to gain further insights into the topic and gain abilities in shell-scripting - and note: “... the only way to really learn scripting is to write scripts” (Advanced Bash-Scripting Guide).

1.3 Further Reading

The following resources provide an introduction to tools and methods considered in this exercise:

- NetCDF: The NetCDF FAQ
The NetCDF Fact Sheet
(<https://www.unidata.ucar.edu/software/netcdf/>)

- CDO: The CDO User's Guide
(<https://code.mpimet.mpg.de/projects/cdo/wiki/Cdo#Documentation>)
The CDO Reference Card
(https://code.mpimet.mpg.de/projects/cdo/embedded/cdo_refcard.pdf)
- Bash: Bash Guide for Beginners
(<http://www.tldp.org/LDP/Bash-Beginners-Guide/html/>)
Advanced Bash-Scripting Guide
(<http://www.tldp.org/LDP/abs/html/>)
- Xarray: Xarray Documentation
(<https://docs.xarray.dev/en/stable/>)

You do not need to read in detail all the information available. Just skim over the text to get a general impression on what the tutorial will be about. Keep the links for reference for more detailed reading in case you need more information during the tutorial.

2 Exercise

2.1 CDO tasks

In the following tasks we will process NetCDF files using the CDO. In order to fulfil these tasks, you may use the documentation (<https://code.zmaw.de/projects/cdo/embedded/index.html>).

First, please download the data files from this link:

https://drive.google.com/drive/folders/1vG-gWFPv-XnufcUZKREFrdeZ2rHJD9w_?usp=sharing

File INIOM_PD_echam5_main_mm_3901-4000_climatological_mean.nc represents the climatological average over 100 years of a recent climate as simulated with a coupled atmosphere-ocean general circulation model. File LGM-W_echam5_6100-6200_climatological_mean.nc represents similar data, but for the Last Glacial Maximum (i.e. the coldest part of the last ice age, about 21,000 years ago). During the Last Glacial Maximum the global average temperatures were lower than today, mostly due to a combination of different insolation conditions and greenhouse gas concentrations. Due to differences in albedo and circulation of the atmosphere with respect to today, some regions were extremely cold.