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:

- **Net**work **C**ommon **D**ata **F**orm (NetCDF): general information on the topic and how to use NetCDF files
- **C**limate **D**ata **O**perators (CDO): several examples of data reduction, analysis and transformation of NetCDF files, including operator piping

In contrast to the commonly used graphical user interface (GUI), most of the work will be done by employing the so-called shell and by formulating commands in a command line interface.

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.

```bash
#!/bin/bash

#select level 6, interpolate to 1x1 degree, and convert from deg. C to Kelvin
cdo timmean -addc,273.15 -remapcon,r360x180 -sellevel,6 input.nc output.nc
#note: the rightmost command is executed first!

#compute global mean
spat_avg=$(cdo output -fldmean output.nc)

#print result to screen
echo
echo
echo "spatial average of global ocean surface temperature is ${spat_avg} K."

#clean up
rm output.nc
```

1.1.3 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).

1.2 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/`)

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 A (subjective) remark: Windows vs. UNIX

For those of you who consider to work in a scientific field with a strong focus on programming or the analysis of large amounts of data: Consider to gain experience with a UNIX-environment, e.g. Ubuntu. In many scientific fields, definitely in climate modelling, UNIX-like operating-systems are the computing environments of choice. Many tools that are necessary for efficiently working in such scientific fields are not natively available on Windows-systems; even if ports are available, they may still suffer from limitations or incompatibilities. It definitely makes sense for you to setup an own Linux partition on your computer (after backuping your data, do not forget that), and to learn how to use and program the available software tools. For every free (free means both "free to use and modify" and "available without a fee") UNIX-environment (e.g. the linux distribution Ubuntu) and the included software tools, a vast amount of detailed – and as well "free" – documentation can be found on-line. There is documentation available for different levels of user experience, addressing both novices and experts.

2.2 Preconditions to solving the exercise problems

For this tutorial it is necessary to run some shell commands with CDO. There are various ways to do that. If you are looking for the least intrusive way on a computer that does not already have a linux system, then using python-cdo for your work is the way to go.

NOTE: IF YOU RECONFIGURE YOUR COMPUTER BY INSTALLING ADDITIONAL OPERATION SYSTEMS OR PERFORMING SIMILAR TASKS, ONLY DO THIS AFTER BACKING UP YOUR DATA TO AVOID INFORMATION LOSS IN CASE OF TECHNICAL PROBLEMS.

- best case: use a linux PC
  it is easy to install the necessary software there (if it is not already available out-of-the-box); commands, how to install software, may vary from distribution to distribution. For Ubuntu, it is as simple as this: open a terminal and hack in the following commands with administrative rights:
  - `sudo apt-get install cdo`
  - `sudo apt-get install ncview`
  - `sudo apt-get install netcdf-bin`

  For Ubuntu there is additional information on setting up your compter provided at `https://paleodyn.uni-bremen.de/study/Dyn2/preparation_NetCDF_CDO_tutorial.pdf`
  If you have a different linux distribution, then it should be easy to adapt installation commands based on information retrieved via a web search with keywords "DISTNAME cdo how to install", replace "DISTNAME" with the name of your linux version
- alternative case: Install Software on MAC
  There is some information available from a previous year's CDO lecture, see:
  `https://paleodyn.uni-bremen.de/study/Dyn2/preparation_NetCDF_CDO_tutorial.pdf`

- NOTE for WINDOWS 10 users:
  Windows 10 comes with the ability to install an ubuntu system as a guest system. Respective documentation is available at:
  `https://ubuntu.com/tutorials/tutorial-ubuntu-on-windows#1-overview`
- There are some other alternative methods on how to run the software for this tutorial. These are outlined in a howto I prepared for a previous year's tutorial:
  `https://paleodyn.uni-bremen.de/study/Dyn2/preparation_NetCDF_CDO_tutorial.pdf`.
  Yet, not all of these methods are possible at the moment.
- In any case, it is also possible to perform all of these tasks purely with python-cdo. The syntax changes only slightly in comparison to the command structure in the shell, but certain outputs require extra steps. This is outlined briefly in the jupyter notebook of the python exercise, but please refer to the documentation for more detailed explanations:
  `https://code.mpimet.mpg.de/attachments/download/27273/python_cdo_introduction.pdf`.

2.3 CDO tasks (4 points)

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.

Identifying the contents and properties of the contained data Consider the file INIOM_PD_echam5_main_mm_3901-4000_climatological_mean.nc:

- Question 1: How many climate variables are present in provided data set?
- Question 2: What temporal resolution does the file have?
- Question 3: How many grid cells (data points) are available per time step and per variable?
- Question 4: What is the physical quantity and unit of the field **snacl**, code 222?
- Question 5: On how many levels is field **q**, code 130, given?

Climatological analysis of two climate states Now we will analyse two different modeled climate states that are representative for the Pre-Industrial (PI, file INIOM_PD_echam5_main_mm_3901-4000_climatological_mean.nc) and the Last Glacial Maximum (LGM, file LGM-W_echam5_6100-6200_climatological_mean.nc):

- Question 6: How much colder is the Northern Hemisphere in January during the LGM if compared to the PI climate?
- Question 8: Q: Which month is wettest (driest) at the location of Bremen (geographical coordinates 53.0793°N, 8.8017°E) for the two climate states if the total precipitation is taken as a measure? Use the operator remapnn in order to interpolate the data to the location of Bremen. Hint: For the LGM, the total precipitation must first be computed by summing up the contributions from convective precipitation (aprc) and large scale precipitation (aprl).
- Question 7: Compute the spatial standard deviation of the annual mean surface temperature (variable **temp2**, code 167) in a rectangular area of 5° × 5° around Bremen (geographical coordinates: 53.0793°N, 8.8017°E). Which climate state has the higher spatial variability?

Notes on submission form of the exercises: *Working in study groups is encouraged, but each student is responsible for his/her own solution. This filled out notebook, together with the answers to the questions from the exercise sheet can be send until the due date (12:00 pm) to Alessandro Gagliardi (Alessandro.Gagliardi@awi.de), Georg Huettner (Georg.Huettner@awi.de).*