1 Overview

In this lecture we will learn to know various helpful tools and techniques that are used in the analysis and processing of climate data. First, we will learn to know the NetCDF file format in a few short practical demonstrations. Second, some data analysis 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
- **B**ourne-**a**gain **sh**ell (Bash): definition of variables, if-then-else construct, error handling, integer-arithmetic, pipes, initializing of shell variables with program output; these programming-methods are illustrated at the example of a simple bash script
- **S**tream **ED**itor (SED): stream processing of strings via variable concatenation

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

In the following some information regarding tools and methods of this exercise is collected in the form of a very general overview. This information collection is intended as a supplement to the lecture. Few further 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.

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 way
- 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, and the sum is time-averaged, 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 ist 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

Shells with their scripting ability are probably the most powerful tool of UNIX-systems, which make 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 the 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
  (`www.unidata.ucar.edu/software/netcdf/docs/faq.html`)
  The NetCDF Fact Sheet
  (`http://www.unidata.ucar.edu/publications/factsheets/current/netcdf_factsheet.pdf`)
- CDO: The CDO User's Guide
  (`https://code.zmaw.de/projects/cdo/embedded/1.6.3/cdo.html`)
  The CDO Reference Card
  (`http://www.iac.ethz.ch/edu/courses/master/modules/radiation_and_climate_change/download/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/`)

2 Exercise

2.1 A 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 sciences, 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 partition on your laptop with a Linux-system, and to learn how to use and program the available software tools. For every free UNIX-environment (e.g. Ubuntu) and the included software tools, a vast amount of detailed, yet free, documentation, addressing both novices and experts, can be found on-line.

2.2 CDO tasks

In the following tasks we will process NetCDF files using the CDO. In order to fulfil these tasks, you may refer to the documentation (https://code.zmaw.de/projects/cdo/embedded/index.html). First, please download the data files for which dropbox links have been distributed.

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.

2.2.1 Identifying the contents and properties of the contained data

Consider the file INIOM_PD_echam5_main_mm_3901-4000_climatological_mean.nc:

- Question 1: What is the physical unit of the field **tsurf**? (1/2 Point)
- Question 2: What is the physical quantity stored in the field **ahfsiac** (1/2 Point)?
- Question 3: On how many levels is the field **t** given? What pressure levels are available? (1 Point)
- Question 4: On how many levels is the field **tsurf** given? Why is there only one level for this quantity? Hint: consider the physical meaning of this field. (1 Point)
- Question 5: How many time steps are contained in the file? (1/2 Point)
- Question 6: What temporal resolution does the file have? (1/2 Point)
- Question 7: How many latitudes and longitudes are considered in the dataset? (1/2 Point)

- Question 8: How many grid cells (data points) are available per time step and per variable (e.g. **tsurf**)? (1/2 Point)
- Question 9: What is the maximum northern latitude? What is the minimum Northern latitude? (1/2 Point)

2.2.2 Climatological analysis of two climate states

In the following we will analyse the Pre-Industrial (PI file INIOM_PD_echam5_main_mm_3901-4000_climatological_mean.nc) and Last Glacial Maximum (LGM, file LGM-W_echam5_6100-6200_climatological_mean.nc) climate states:

- Question 10: What is the global average annual average surface temperature (variable **tsurf**) for PI? (1/2 Point)
- Question 11: What is the global average annual average surface temperature (variable **tsurf**) for LGM? (1/2 Point)
- Question 12: How much colder is the northern hemisphere in January in the LGM if compared to the PI? (1/2 Point)
- Question 13: Which month is warmest in the LGM (northern hemisphere, southern hemisphere, global mean)? (1/2 Point)
  Bonus question (will earn you one additional point in case you miss at least one point from the other questions): Why is the warmest month in the global average the same as for the northern hemisphere?
- Question 14: Which month is warmest in the PI (northern hemisphere, southern hemisphere, global mean)? (1/2 Point)
- Question 15: What are the following climatological characteristics at the location of Bremen (geographical coordinates 53.0793°N, 8.8017°E for both climate states (use the interpolation operator remapnn): annual minimum surface temperature (variable **tsurf**), annual maximum surface temperature, annual mean surface temperature. (1 Point)
- Question 16: If you consider the annual cycle of surface temperature: What is the difference between warmest and coldest month in the northern hemisphere for each of the two climate states? (1 Point)