Software Package for Climate Modelling Diagnostics

ADANNA AKWATAGHIBE, Imperial College London
DR. YVES PLANCHEREL*, Grantham Institute, Imperial College London

1 INTRODUCTION

Climate models are a core component of climate science and they help climate modellers to predict future climate and understand how human influence is affecting the Earth's climate [13]. There is increasing interest in predicting future climate not just by climate modellers but also by insurance companies, energy companies, hedge funds, farming businesses, foreign aid - individuals that are not traditionally involved in climate modelling, but will either benefit or suffer from climate change. There has been a democratisation and internationalisation of the climate modelling effort. Climate model output is becoming more accessible, especially with cloud computing and more high-performance computing (HPC) systems become increasingly available in companies and universities. The data output of the models are usually stored around the world in different distribution centres, so the data is accessible from all over the globe [9].

Since the first climate model in 1994 [20], the number of climate models, their size and the number of numerical experiments to test climate scenarios, model performance and sensitivities have been increasing rapidly. As more and more models are being run, the spatial and temporal resolution of the grids underlying the structure of these models have also been improving.

Computers that are large enough to run 3-dimensional climate models use tens to thousands of processors and are GPU-accelerated. There has been an improvement in computational power which enables us to run these models and their ensemble simulations (i.e. performing similar experiments that either start from various initial conditions or use different parameter values) [20].

Additionally, the frequency of the data has changed. The output of the model was yearly or monthly, however climate modellers have started to take an interest in more high frequency data - daily iterations of data. This is because the daily values of the climate model output affects averaging as well as the rectification (i.e. bias correction) of the data, and potentially gives more accurate values of these calculations [17].

The initial and boundary conditions that are needed to run the models affect the forecast predictions of the climate, but we still need to be able to get the statistical data to predict the future climate. This is why it is imperative for us to understand the link between low and high frequency data outputs.

The model output has increased dramatically in combination with the amount of model simulations available, it is thus important to develop tools to handle and interpret these data in an efficient and reproducible manner.

1.1 Objectives

The objective is to develop computationally efficient functions that:

- Calculate climate model diagnostics that can be used to understand climate patterns better.
- Create output for the user that effectively summarises the data in a useful way by providing diagnostics capabilities. The user will have choices of diagnostics e.g. mean, standard deviation, Net Primary Production (NPP). The user can also input their own function to use as a diagnostic.

*Supervisor of Independent Research Project (IRP)

- Engage with the climate modelling community to try and understand and isolate the common needs of the modellers.
- Create a software package and contribute to existing software like **Iris** [23] or **xarray** [19]. The package will contain routines that do not already exist in these software.

2 CLIMATE MODELS

Firstly, it is necessary for us to distinguish the difference between predicting day-to-day weather, or *Weather forecasting* and Climate Modelling.

Weather forecasting tells us how the weather will be and the specific temperature within a *short* time period. On the other hand, Climate is a *statistic* description of the state and variability of a system, and gives us the *projections* of weather that occur in the far future (years or decades ahead) [25]. As an example, a climate model can predict that it will be warm in the summer of 2070 but it cannot tell you the exact temperature that it will be on a specific day. However, weather forecasting can tell you that tomorrow it will be $20°C$ or even more specifically, at 2pm tomorrow, it will be $20°C$.

Weather forecasting and climate modelling forecast predictions should theoretically be the same, but since we do not have infinite knowledge about the future, this is not practically possible. That being said, the UK Met Office have developed a climate analysis tool called the UK Climate Projections 18 (UKCP18), and it provides users with the most recent scientific evidence on climate change projections that they can use to plan for the future [11]. UKCP18 includes estimates of the range of probable outcomes of future climate in the UK. Therefore, the tool does not just use weather forecasting to predict daily or weekly temperature but it is using climate model outputs to predict and prepare for future climate change. The UKCP18 tool is a good indication of the push towards long-scale weather forecasting in order to help decision makers assess their risk exposure to climate [11].

Climate change refers to a change in the statistical description of the mean state of the climate or its variability. This change must last for an extended period of time [26].

2.1 What are climate models?

Climate models, or Coupled Global Circulation Models (CGCMs) are designed to represent the Earth's climate system. These are tools used to further our understanding of the Earth's past, present and future climate projections [14]. The *primary earth system components* that the coupled or Earth system model simulates include ocean, atmosphere, ice sheets and land. The models are run with many simulations under different scenarios to investigate the effect of each scenario in future climate conditions. One scenario could be running a model with only natural forcing (e.g. taking volcanic eruptions, solar output etc. into account) and another would be running the model with both natural forcing and human influences (also taking into account the emission of carbon and greenhouse gases or the effect of aerosols on the atmosphere). The model outputs are then measured against observed real-life data.

Figure 1 (a) shows that with natural forcing and human influence, the model fits closely to the observed data in the global mean temperature. Conversely in (b) without the inclusion of human forcing, the fit between the observed data and the model diverges towards the end of the 20th century.

In Figure 1, the models are compared with observed data, but this may not always be available, so the model needs to be compared to what is called a *control run* or *control simulation.* A control run is also needed to study how biased the run is compared to the actual data. Control runs are

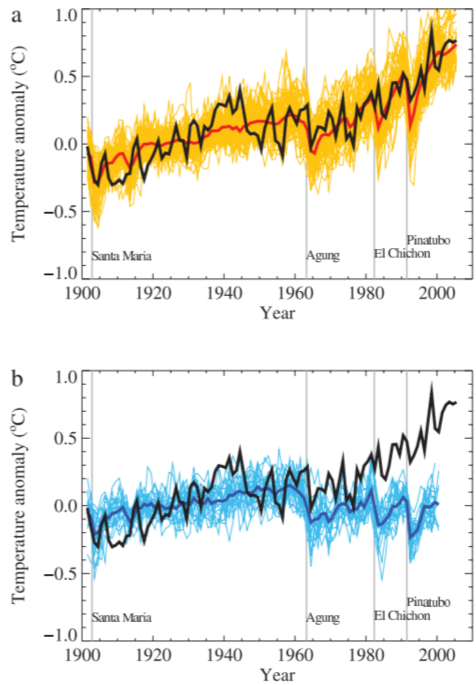

Fig. 1. Comparison between global mean surface temperature anomalies (°C) from observations (black) and AOGCM simulations [18].

used to establish a baseline for the model. By comparing the climate model experiment with its control run, the climate change signal of the model can be detected. Similarly, the climate change signal of the observed data can be detected by comparing the observed data with the control run.

Since the model experiment may drift away from the observed data, the bias between the control run and observed data may dominate the climate change signal of the observed data. Since the control run is defined as a baseline of the model, the bias between the control and the model experiment should be 'cancelled', or at least minor compared to the climate change signal of the model [21].

The initial conditions (greenhouse gases, solar activity etc.) of the model are held in a reference state. Pre-industrial values of the initial conditions are used and the model is run for hundreds or thousands of years until it reaches a stable *dynamic* equilibrium, that is, the model output more or less has a specific pattern over hundreds of years. Control runs can be used to see and understand the *natural variability* of the climate - when multiple simulations are run with the same model with different initial conditions and the same (natural) forcing, the model output will vary for each simulation [18]. Natural variability is influenced by both natural variations (solar activity, volcanic eruptions) and also by internal variations, which are developed from coupled interactions between the earth system components, such as those happening in the tropical Pacific Ocean during an El Niño event (ENSO) [18]. These natural variations produces seasonal, yearly or even decades-long fluctuations in the climate.

2.2 How are CGCMs built?

CGCMs divide the primary earth system components into a 3-dimensional grid space and discretise and solve the equations governing the components using supercomputers, as these are extremely large systems to solve. Since the model is coupled, the equations are also solved using multiple processors in order to streamline and speed up the communications between components [20].

Each of the Earth system components are submodels which need to be connected together to form a coupled model. This interconnection is achieved using a *model coupler*. The submodels use differing grids and resolutions from each other, which means that their modelling approaches will be different and will have to be dealt with by the model coupler [20]. The model coupler controls the execution and time evolution of the climate model by coordinating and controlling the passage of information between the components: atmosphere, land, ice sheets and oceans [20]. Figure 2 shows the model coupler between the components, and illustrates the transfer of the fluxes from one component to another.

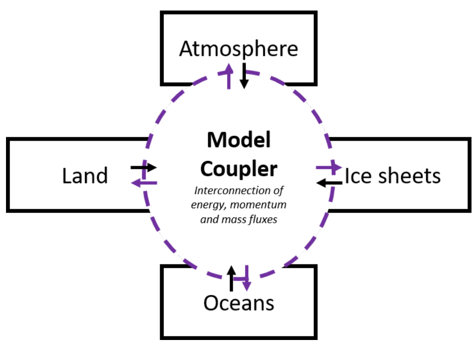

Fig. 2. Model coupler

There are different kinds of grids used to discretise the system of equations.

2.2.1 Finite grids. *Finite grid discretisation* is done by dividing the subsystems of the climate into rectangular/square grids. This is also known as the latitude-longitude grid. This was the earliest discretisation technique for both the ocean and the atmosphere [20]. The main disadvantage of this technique is that it suffers from *the polar problem*, where at the poles, the grid spacing becomes very small and converges to a point as shown in Figure 3a. The reduction of the spacing means that in order to maintain computational stability near the poles, smaller and smaller time steps have to be used. Unfortunately, this means that the computational efficiency would be reduced. Many models solve this stability and inefficiency problem by using *semi-Lagrangian timestepping* [20].

2.2.2 Spectral models. Many *ocean* models use finite grid discretisation, and a large number of *atmospheric* models use *spectral models* [20]. These models make use of the the spherical geometry of the earth when calculating the equations. These models use finite volume and finite element methods to solve the system of equations [28]. In order to solve the equations, we need to transfer from grid space to spectral space and vice versa. This is actually the main issue with using this method as most of the computational time lies in the transformations between grid and spectral space at each timestep. However, unlike with finite grids, spectral models do not have the *polar problem*, they are also extremely efficient and accurate. Also, the structure of the grid allows the possibility of increasing and decreasing the resolution on regions of higher and lower interests respectively.

(a) Finite grid points (b) Geodesic grid

Fig. 3. Different types of discretisation grids

2.2.3 Geodesic grids. Geodesic grids aim to keep the advantages of the finite grids and spectral models while reducing the effect of their disadvantages when solving the equations. The grid covers the sphere of the earth with triangular grids. Figure 3b shows a diagram of a geodesic grid. There are two main advantages of using this kind of grid, the first being that since all the grid cells are nearly the same size, the uniformity allows for computational stability. The second advantage is its ability to be parallelizable, given that the grid can be split into neighbouring blocks each of which can be solved using a separate processor. Similarly to spectral models, the resolution on regions of higher and lower interests can be increased or decreased respectively.

In our software package, it may be necessary to regrid the data depending on the representation of the climate model data given by the user. If the model output is obtained from the *Program for Climate Model Diagnostics and Intercomparison* (PCMDI) [9], then the output is likely to be represented using a latitude-longitude grid, and can be used in our code without any adjustments. Otherwise, if the model data is obtained directly from a modelling centre, then the grid is highly unlikely to be rectangular, and may be geodesic. This means that regridding techniques will need to be used to interpolate data from geodesic grids onto latitude-longitude grids.

2.3 Brief history of CGCMs

At the start of the production of GCM models not coupled and were either specifically atmosphere model or ocean models. Over time, GCMs became coupled atmosphere-ocean general circulation models (or AOGCMs). Nowadays, more components of the earth system (e.g. Ice sheets, carbon cycle etc.) are added to the CGCMs [27]. *Earth System* Models (ESM) are CGCMs that include additional information on biogeochemical cycles [27]. The data obtained from ESMs will be used in our software. The Intergovernmental Panel of Climate Change (IPCC) was established by World Meteorological Organisation (WMO) and the United Nations Environment Programme (UNEP). Their objective is to produce the most recent developments in climate science. Synthesis or assessment reports are published every five to seven years. The most recent is the AR5.

Over several years, several climate models have been run in order to give outputs that are vital and included in the IPCC assessment reports. The most recent models were run for AR5 from the Coupled Model Intercomparison Project 5 (CMIP5). The model experiments from CMIP6 are currently being run to be injected into the new assessment report AR6 which is in development.

CMIP was developed as a means standardise the model experiments so that the setup for each model would be the same. This means that now if the model experiments are setup the same and their outputs differ, we know it is because of the model processes as opposed to the experimental design [13], i.e if the models were run with the same initial and boundary conditions, certain causes for the differing model output can be eliminated. The intercomparison project created a coordinated global community-wide experiment. And, it allows each modelling centre to conduct model experiments that transcends their own models capabilities, by having other modelling groups perform experiments with the same setup on their own model (that may have more capabilities).

CMIP goes through a new iteration each five or six years. In each iteration, all the modelling centres are given a specific set of experiments to run. In earlier iterations, CMIP experiments would be to model the impact of a 1% annual increase in atmospheric CO2 emissions and analyse between 1-1 models. More recent CMIP experiments have increased the emission scenarios and their detail as well as increasing the number of models that are being compared [9]. In the IPCC assessment report, the analyses of the model output (from articles written by the scientists running the models in the modelling centres) are summarised. The *Program for Climate Model Diagnostics and Intercomparison* (PCMDI) is an organisation that provides software and storage for the CMIP data in distribution centres globally. PCMDI deals with the diagnosis and intercomparison of CGCMs [9].

Figure 4 shows the increase of the number of models, resolution, model processes, variables, computing power and data as the CMIP experiments evolve. Historically, the resources needed to run a model was not so computationally intensive, but with the increase of all these attributes and resources, nowadays the need for computational efficiency when running these models and when dealing with their output is highlighted and this is something we need to keep in mind when implementing our software.

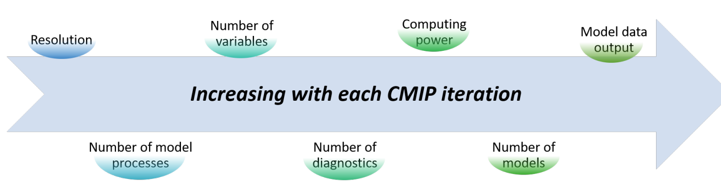

Fig. 4. CMIP models evolution

2.4 Ensembles

An *ensemble* of a model involves different variations of the same model, run with either with a change of initial conditions or parameter values. Ensembles are used to understand the uncertainty of the climate simulations. There are two types of ensembles:

2.4.1 Multi-model ensembles. Multi-model ensembles or equivalently, *ensembles of opportunity* explore the *structural uncertainty* of a models. The structural uncertainty is present when we are uncertain about the model output because we are unsure about the different numerical techniques or resolution of the model. These uncertainties are limited in how they are used because the model structure cannot be changed [20].

2.4.2 Perturbed physics ensembles. In a *Perturbed physics ensemble*, the parameter values of the models in the ensemble are changed so they differ from one another. The models are run and their outputs shows the effect of the parameter values [20]. This particular ensemble is used to show the *representational uncertainty* - the uncertainty we get from changing the parameter values.

The average of an ensemble gives a good indication of the direction of change. The standard deviation of the ensemble gives an indication of the robustness of the models [22].

2.5 Climate model data: NetCDF files

The output of the climate models are usually saved as NetCDF files. NetCDF or *The Network Common Data form* is library that deals with the storage and retrieval of data in arrays. NetCDF files are machine-independent datasets and the data from the files can be extracted or created using the NetCDF libraries [7]. NetCDF have different formats of files, all of which offer different features.

The data saved in a NetCDF files are saved in a container called a `dataset`. This dataset contains:

- `dimensions`
- `variables` , each variables has:
  - `attributes`
- `global attributes`
  - `description`
  - `history`
  - `title`
  - `file name` etc.

  etc.

2.6 Climate model users

As mentioned before, there are many individuals who are not climate modellers who are interested in predicting future climate e.g. UN, insurance companies, farming businesses etc. However, these potential users of climate models may have limited experience with climate models, while many of the users who access model outputs from the storage capabilities of PCMDI are climate modellers.

Therefore, creating a software package that caters to both climate modellers and those with limited experience to climate modelling, and having both groups of users as operational users of our software would increase the scope of utility of climate models. And the hope is that those who have not yet been exposed to climate models will find our software to be very useful and a step in the right direction.

3 EXISTING SOFTWARE

3.1 Iris

Iris is specifically catered to climate modellers or at least those with some level of experience in the field. Iris is a Python library that has a "powerful, format-agnostic interface" for working with Earth Science data. It contains functions able to merge, convert, **visualise** (using Matplotlib [3] and Cartopy [2]), **aggregate**, reduce, **interpolate** and **regrid** the data amongst other things [23].

The library is able to work with many different file formats, including NetCDF. It also builds on Numpy [8] and dask [5] to scale for both single CPU systems and for multiple processors [23].

The functionality highlighted in bold will be useful to us when we use Iris in our project. Aggregating data (or combining data within the time or spatial axis) is useful when calculating the ensemble means. Interpolation is necessary when calculating a value of the variable at a sample point in the latitude-longitude grid. Regridding the data (converting grid used in data from one grid to another) is necessary if we want to convert the Cartesian coordinates to spherical coordinates, since we are dealing with points on the Earth. It is also useful for converting from geodesic grids to latitude-longitude grids.

3.2 xarray

xarray is a more recently developed Python package that focuses on working with labelled multi-dimensional arrays in a simple and more straight-forward manner. The package is tailored to work with NetCDF files and deals with reading and writing of data, interpolation, data arithmetic, time series and more. Inspired from pandas [4], it also uses dask [5] for scalability [19].

Since xarray is specialised in multi-dimensional arrays and NetCDF files, most of the functions in the package will be extremely useful to incorporate in our code.

4 OUR SOFTWARE PACKAGE

Neither Iris or xarray provide the user diagnostics or summaries that our software will provide. Also using these packages is quite involved, the user needs to know python and know how to call and use functions in python. For our software, the aim is to cater to both users with experience with python and also those without. Currently users need to know how to read NetCDF files and pre-process their data. This means having to work with several software in each step [27]. Our package combines all of this into one.

The user is expected to go through these basic steps in our software:

```
Input data -> Select options -> Save data  + Give summary
```

Notice how minimal these steps are, the user does not need to know how to program in Python, although if they do have that experience, in the `Select options` phase, they can input their own python code for us to run as a diagnostic. A more detailed chart will be shown in the *High level design* section.

4.1 Aims to accomplish in project

Note that in the list below: `FP` - Functionality is included in the first prototype, which can be found in our project plan git repository [6].

---

(1) **ENTRY:** Takes in user input (`FP`)
The requirement arguments needed in the user input are prefix of file names, start date of analysis, variables and the number of ensembles. The optional arguments are end date of analysis, plotting option, monthly option (the model output files are assumed to use daily data by default), grid point, sample point, file that contains mask for grid, save output option, co-variance option and histogram bin selector (how we calculate the number of optimal bins to use in the histogram, the default is the Freedman Diaconis Estimator [15]. The user can input these arguments in two ways:
- from command line
- from file input - stored as a text file that has the list of all the arguments for the user to fill in. It may be clearer/easier to use for users unfamiliar with the command line interface

(2) **EXTRACT:** Using user input, extract variable data from netCDF files given by user
- 2D rectangular grid (`FP`) or geodesic grid.
- masks - a list of polygons (with their vertices given) that will overlay the 2D grid and separate data into different regions. Figure 5 shows an illustration of the masks overlay.
- grid point - select the grid in the 2D grid that corresponds to the latitude and longitude given, or the grid that the point falls in. (`FP`)

- sample point - calculate the variable value of the exact point of latitude and longitude using interpolation. (FP)

(3) **ANALYSIS:** Calculate diagnostics
- compute average of ensembles (FP)
- compute standard deviation of ensembles
- histogram
- timeseries
- covariance of variables (e.g. 2D histogram)
- user defined diagnostic functions to pass into software
- merging code used to calculate the ENSO Indices, Indian Ocean Dipole (IOD) Mode Index, Atlantic Multidecadal Oscillation (AMO) Index, Pacific Decadal Oscillation (PDO) Index, Arctic Oscillation (AO; Northern Annular Mode) and Antarctic Oscillation (AAO; Southern Annular Mode) Index and the North Atlantic Oscillation (NAO) Index [16]. The (command line) code is provided by **Davor Dundovic**, an undergraduate student in the Department of Earth Science and Engineering at Imperial College London. His code will be re-written in Python and added to the list of functions that the user can call.
- more user diagnostics to decide

(4) **SAVE OUTPUT:** Save output of diagnostics
- save to netCDF files for each ensemble in a folder that holds all the ensembles means. (FP)
- save to `.dat` files, a generic data file that can store tables that are easy to read in python.

(5) **PARALLEL:** Parallelisation when calculating diagnostics
- splitting the calculations of models from the ensembles to different processors using *Message Passing Interface* (MPI) [12].

(6) **SUMMARY:** Creating a summary of output that is useful for the user
- summarise change variable with respect to time e.g. by 2050, SST will be greater by 10%.
- interpret timeseries / histogram
  - Calculate range of change of variable over length of time
  - frequency of low and high events

(7) **USER:** Providing a software for operational users. Not developing the software package in isolation but contributing and engaging with the climate modelling community. In the *Testing* section, we give an example diagnostic that was suggested to us by a climate modeller according to his scientific interest.

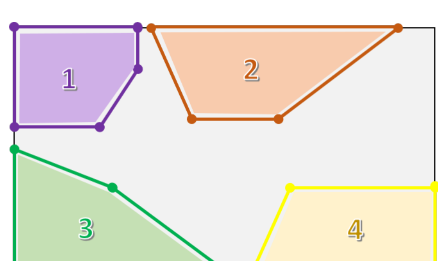

Fig. 5. Polygons 1, 2, 3 and 4 are overlayed on the 2D grid. Vertices of polygons are represented by circles.

4.1.1 Problems still to solve.

(1) Masks (illustration is shown in Figure 5)

    - given a list of polygons, isolate data from different regions, possibly using the *Generic Mapping tools* (gmt) [1].
(2) Interpolation problems when extracting data with a sample point
    - masking regions and doing interpolation in the specific regions so that they do not contaminate each other e.g. there are some small islands which divide the Indian and Pacific ocean. To get a sample point in the Indian ocean, we do not want to interpolate using points in the Pacific ocean. The data points in Pacific and Indian ocean should not rely on each other. Another example is the Panama Canal, an artificial waterway that connects the Atlantic and the Pacific Ocean. In this case, getting a sample point on the boundary of the Pacific should *not* use interpolated points from the Atlantic, and vice-versa.
    - use a more accurate interpolation function in gmt `greenspline` instead of `interp2D` in `scipy`, because `interp2D` ignores the spatial coordinates of the grid points.
    - Geographic distance between points is not the same as latitude/longitude distance. The points up in the poles are closer than around the middle.
    - Wrap around interpolation problem since Earth is spherical.
(3) Expand function `extract_data` to be suitable for 4D data

4.2 Implementation

4.2.1 High level Design. Our software is split into five sections:

(1) User input
(2) Extract variables and time period from data
(3) Calculate diagnostics
(4) Save output from diagnostics
(5) Summarise the data in a useful way

The high level diagram is shown in Figure 6.

4.3 Testing

Unit tests and Integration tests will be written using Pytest framework. Some unit tests have already been written in `main_tests.py`. Continuous Integration (CI), specifically Travis CI [10] is setup on our git repository, and each time the code is pushed to git, our unit and integration tests are run to check if any new changes to the code cause any problems, and if they do, then the errors can be detected quickly.

4.3.1 User testing. We plan on doing user testing with some climate modellers and also those who have less experience with climate modelling.

Based on user requirements, Net Primary Production (NPP) will be used as an example of a diagnostic. NPP is which is how much carbon dioxide plants takes in during photosynthesis minus how much carbon dioxide the plants release during respiration [24].

The NPP diagnostic was added based on the needs of by **Dr. Keith B. Rodgers** in the new modelling centre IBS Center for Climate Physics in Korea. He also provided the climate model output that we will test with, the data is oceanographic data of different algae types.

Dr. Emma Cavan a research associate in Ecosystem Modelling, in the Department of Life Sciences at Imperial College London will be a tester as a non-climate modeller.

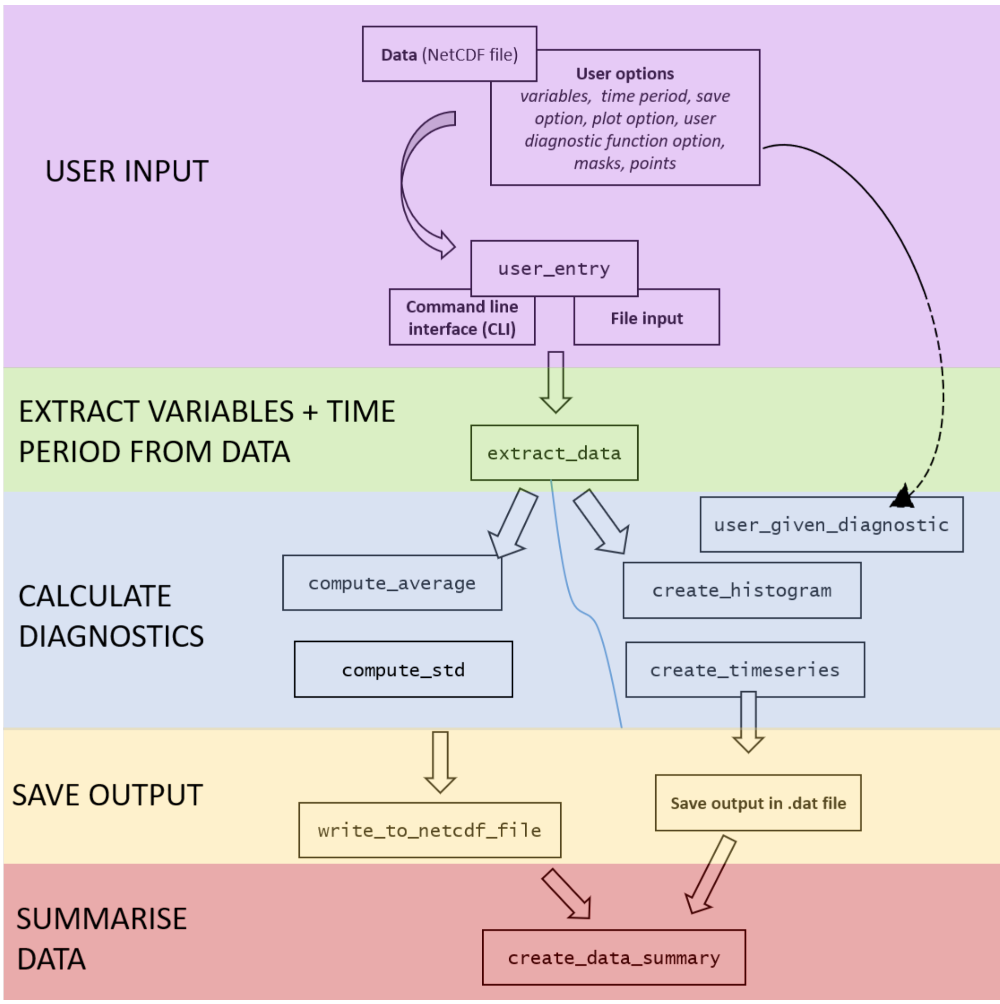

Fig. 6. High level structure of code

5 OUTLINE OF FINAL REPORT

The final report will be structured in a very similar way to the project plan

- Abstract
- Introduction
  - Objectives
- Climate models
  - What are climate models?
  - How are CGCMs solved?
  - Brief history of CGCMs
  - Ensembles
  - Climate model data: NetCDF files
  - Climate model users
- Existing software

  - Iris
  - xarray
- Software package
  - Requirements analysis
  - High level Design
  - Implementation
    - Input of software
    - Output of software
  - Analysis
    - Functionality of:
      - `compute_average`
      - `compute_std`
      - `create_histogram`
      - `create_timeseries`
      - each of the indices calculated: ENSO, IOD, AMO, PDO, AO, NAO
      - user given diagnostic
  - Testing
    - Unit tests
    - Integration tests
    - User testing
- Evaluation of software
- Conclusions

REFERENCES

[1] 2013. Generic Mapping Tools, Overview. Retrieved June 24, 2019 from http://gmt.soest.hawaii.edu/projects/gmt
[2] 2018. Cartopy. Retrieved June 24, 2019 from https://scitools.org.uk/cartopy/docs/latest/
[3] 2018. Matplotlib. Retrieved June 24, 2019 from https://matplotlib.org/
[4] 2018. pandas. Retrieved June 24, 2019 from http://pandas.pydata.org/
[5] 2019. dask. Retrieved June 24, 2019 from https://dask.org/
[6] 2019. Github Project Plan repository. Retrieved June 27, 2019 from https://github.com/msc-acse/acse-9-project-plan-AdannaAkwats
[7] 2019. An Introduction to NetCDF. Retrieved June 24, 2019 from https://www.unidata.ucar.edu/software/netcdf/docs/netcdf_introduction.html
[8] 2019. Numpy. Retrieved June 24, 2019 from https://www.numpy.org/
[9] 2019. Program for Climate Model Diagnosis and Intercomparison. Retrieved June 24, 2019 from https://pcmdi.llnl.gov/index.html
[10] 2019. Travis CI. Retrieved June 27, 2019 from https://travis-ci.org/
[11] 2019. UK Climate Projections. Retrieved June 27, 2019 from https://www.metoffice.gov.uk/research/collaboration/ukcp
[12] Blaise Barney. 2019. Message Passing Interface (MPI). Retrieved June 28, 2019 from https://computing.llnl.gov/tutorials/mpi/
[13] Carbon Brief. 2018. How do climate models work? Retrieved June 24, 2019 from https://www.carbonbrief.org/qa-how-do-climate-models-work
[14] J. Marotzke B. Abiodun P. Braconnot-S.C. Chou W. Collins P. Cox F. Driouech S. Emori V. Eyring C. Forest P. Gleckler E. Guilyardi C. Jakob V. Kattsov C. Reason Flato, G. and M. Rummukainen. 2013. Evaluation of Climate Models. In: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. (2013). https://www.ipcc.ch/site/assets/uploads/2018/02/WG1AR5_Chapter09_FINAL.pdf
[15] P Freedman, D. & Diaconis. 1981. On the Histogram as a Density Estimator: L_2 Theory. *Z. Wahrscheinlichkeitstheorie verw Gebiete* 57 (1981), 453–476. https://doi.org/10.1007/BF01025868
[16] AOPC/OOPC GCOS. 2019. Working Group on Surgace Pressure. Retrieved June 27, 2019 from https://www.esrl.noaa.gov/psd/gcos_wgsp/Timeseries/

[17] Hagemann S. Moseley C. & Piani-C. Härter, J. 2011. Climate model bias correction and the role of timescales. *Hydrology and Earth System Sciences* 15 (2011), 1065–1073. https://doi.org/10.5194/hess-15-1065-2011

[18] F. W. Zwiers P. Braconnot N.P. Gillett Y. Luo J.A. Marengo Orsini N. Nicholls J.E. Penner Hegerl, G.C. and P.A. Stott. 2007. Understanding and Attributing Climate Change. In: Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change. (2007). https://www.geos.ed.ac.uk/homes/ghegerl/AR4WG1_Pub_Ch09.pdf

[19] S. Hoyer and J. Hamman. 2017. xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software* 5, 1 (2017). https://doi.org/10.5334/jors.148

[20] Ann Henderson-Sellers Kendal McGuffie. 2014. *The Climate Modelling Primer* (4th ed.). John Wiley & Sons, Inc.

[21] Christian Kerkhoff, Hans R. Künsch, and Christoph Schär. 2014. Assessment of Bias Assumptions for Climate Models. *Journal of Climate* 27, 17 (2014), 6799–6818. https://doi.org/10.1175/JCLI-D-13-00716.1

[22] Sedláček Jan Knutti Reto. 2012. Robustness and uncertainties in the new CMIP5 climate model projections. *Nature Climate Change* 3 (2012). https://doi.org/10.1038/nclimate1716

[23] Met Office. 2010 - 2013. *Iris: A Python library for analysing and visualising meteorological and oceanographic data sets* (v1.2 ed.). Exeter, Devon. http://scitools.org.uk/

[24] NASA Earth Observatory. 2019. Net Primary Production. Retrieved June 27, 2019 from https://earthobservatory.nasa.gov/global-maps/MOD17A2_M_PSN

[25] Australian Academy of Science. 2019. What is Climate change? Retrieved June 24, 2019 from https://www.science.org.au/learning/general-audience/science-climate-change/1-what-is-climate-change

[26] World Meteorological Organization. 2019. Frequently Asked Questions (FAQs). Retrieved June 24, 2019 from http://www.wmo.int/pages/prog/wcp/ccl/faqs.php

[27] Sara Rivero-Calle Rebecca G. Asch, Darren J. Pilcher and Johnna M. Holding. 2016. Demystifying Models: Answers to Ten Common Questions That Ecologists Have About Earth System Models. *Association for the Sciences of Limnology and Oceanography* (August 2016), 241–269. https://doi.org/10.1002/lob.10113

[28] Paul Aaron Ullrich. 2011. *Atmospheric Modeling with High-Order Finite-Volume Methods.* Ph.D. Dissertation.