Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project

Project Plan

EZclim2.0 - A software package for Climate Modelling diagnostics

by

Yusen Zhou

yz219@imperial.ac.uk

GitHub login: acse-yz219

Supervisors:

Dr. Plancherel Yves

June 2020

GitHub repository: https://github.com/acse-2019/irp-acse-yz219

Introduction

Modeling on the coupled climate system is very important, it helps us to understand the multi-scale dynamic interactions between natural and social systems that affect climate [1]. However, questions related to this area are usually too large and too complex to be solved by a single team, scientific agency, or nation. Thus, the World Climate Research Programme (WCRP) formed a working group of Coupled Modelling (WGCM) and start it's Coupled Model Intercomparison Project (CMIP)[2]. Since 1995, CMIP has attracted multiple international modeling teams worldwide to contribute to the climate model experiments. CMIP defines experiment protocols, forcing, and output. These outputs can be retrieved from the Centre of Environmental Data Analysis (CEDA) which hosts over 13 petabytes of atmospheric and earth observation data. CEDA also provides Jasmin service which is the global unique data-intensive supercomputer for environmental science. Using Jasmin, scientists can access curated data in the CEDA Archive and use the power of the supercomputer to accelerate their research. With the CMIP come to its sixth phase, over 70 models participated in the model group and a large proportion of them are newly added models[1]. These models are often more complex and have higher resolution. Thus, how to analyze the model output in an efficient way become a big problem. In this case, Adanna Akwataghibe from ACSE-2018 developed a new software package called EZclim last year. EZclim was developed using Python and combine the step of reading, pre-processing, and calculation of climate modeling diagnostics into a singular program. Climate modeling diagnostics like the mean, median, standard deviation, and root mean squared error can be calculated by EZclim and all these outputs are saved in NetCDF files which can be represented visually using maps, histograms, time series, and box plots. Besides, the software has the ability to re-grid, extract regions, calculate climate indices, rotate the grid among other functionalities. However, due to the time limitation, this software package still has a lot of details to be filled. Firstly, all the output files EZClim1.0 needed should be downloaded first. Considering the limitation on PC storage, it will be better if the data can be read directly using Jasmin service. Secondly, scientists sometimes want to combine data from different models but EZClim1.0 can only support single model diagnostics [3]. Thirdly, the external user function interface of EZClim1.0 was hardcoded. It is not flexible enough to adapt to the different external user functions. Fourthly, EZClim1.0 has no graphical user interface so it is hard for users that don't familiar with python or Linux operation. Besides, several objectives of last year are not achieved completely like resolution problem. According to the above, the EZclim2.0 version will be developed in this project to fix these problems.

Project Plan

Objectives

The basic objective of this project is to read different data from different model outputs and combine them together to realize external user functions. The previous interface of EZclim1.0 will be softcoded to adapt to different external user functions. After finishing this, possible researches will be carried on these user functions like applying the machine learning method or developing plot function. The optional objectives will be to finish the unachieved objectives from last year or design a GUI for the EZClim tool.

Implementation

To read different model outputs correctly and simultaneously on Jasmin. The model outputs should be stored in a format following the rule of the CEDA. The storage rule should be figured out to resolve the data address and get access to the data. The data retrieve process might be done using the newly developed evaluation tools on jasmine called ESMValtool. EZClim only needs to realize the interface with it. However, if the ESMValtool doesn't work well, this function will be achieved by EZClim itself. And for integrate different user functions into EZclim. EZclim1.0 already has the code to realize the integration but it cannot combine data from different models. However, sometimes users need to use different model outputs to do analysis. Thus, the code needs to be changed into reading different variables from different files. To achieve this object, list data for each variable will be read separately from the data files and merge together into an iris cube. This iris cube will be passed to the user function for later use. External user functions often take list type climate variables data as input, but in this project, all the user functions should be written into a format that taken cube data as input, and all the parameter information will be read from the cube in the format like cube['temperature']. In this way, the other part of the external user function doesn't need to be changed.

After finishing the basic part, several unachieved objectives remained from last year will be tried. The first is the resolution improvement. The highest resolution EZclim1.0 used is daily data, but in some cases, hourly data will be required. The problem with using hourly data is it will take a long time to run the code. To solve this, the parallelization approach will be extended to deal with high-frequency data output. The second is conserving space when extracting a region. Currently, when extracting a specific region with a mask, EZclim1.0 set all elements outside the masked region to NaN and doesn't change the grid size of the original data. In order to conserve space, in EZclim2.0, the original grid size will be changed to that of the masked region. Besides, Interpolation around the edges of the mask for any non-regular polygons would have to be implemented. The third objective will be solving possible memory limitations when writing files. EZclim1.0 writing the final output of EZclim in a NetCDF format file, this file will often be combined with the original data file as output. However, for large data files, the writing time and memory usage will have a significant increment. EZclim2.0 will check the memory usage of the software while it is running and then decide based on the CPU usage and write-memory available. This will be used as a judgment of when to save a combination of files or just the bare minimum needs to be investigated.

When the project comes to the final stage, if it still has some time remaining before September, a graphical interface will be developed. For climate modelers that are more comfortable with using the command line and for those people who are advance in python, the current interface design for EZClim is already good enough. They can either give input by editing the input file or directly input in the command line. However, the graphical interface should be a supplementary option for people that are non-climate modeler or not familiar with Python or Linux operation. Users can input parameters like start date and end date on the developed graphic interface instead of editing the input files which will be more convenient. There is another advantage for this, the user interface will check if the input parameter is valid or not, so users don't need to wait until the program run to know if the input is correct or not. Besides, users can select the user function from the drop-down list instead of writing it on the input file. After running the program, users can decide whether to plot the histogram or not. All the GUI function above will be developed using PyQt. PyQt is a tool package for creating python GUI which is developed by Phil Thompson.

Test

All the stages above will contain several unit tests to ensure that all the functions are achieved successfully, and the new features will not affect the overall program. This including tests for reading data, each external function, and other functionality changes.

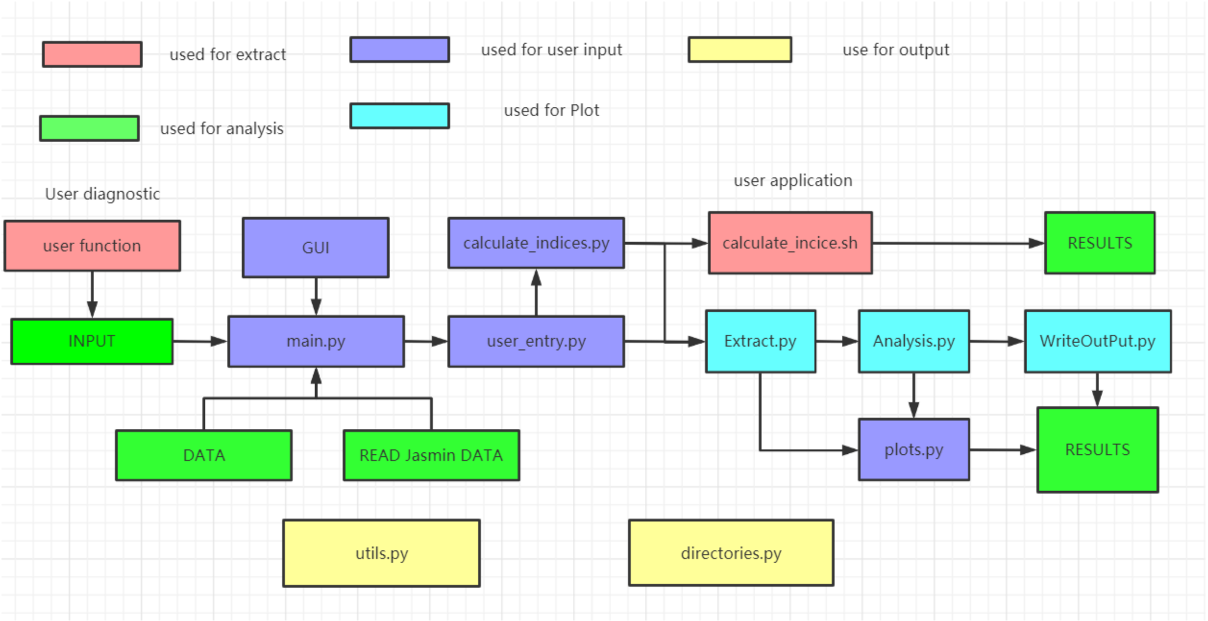
Besides, an integrated test will also be carried out to test the overall working flow and functionality. Some researchers with experience in related fields will be invited as a part of the integrated test. They will check if the functionalities meet their requirements and whether the software is user-friendly.

Risk management

Risk management is very important in this project since all the work should be done on Jasmin. Jasmin service is provided by CEDA, it usually works well. However, sometimes its server will crash for some reason like someone took it to mine Bitcoin. The backup plan will be to download data directly from the CEDA webpage and run the program locally and attention will be more focused on the user functions part.

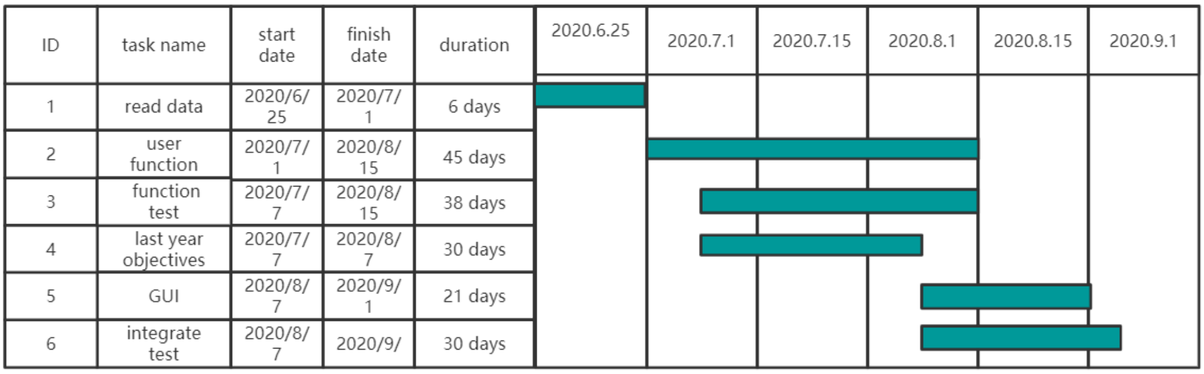
Software Structure

The structure of the software is shown below.



Gantt chart

Gantt chart is used here as a timeline for the project



ID	task name	start date	finish date	duration	2020.6.25	2020.7.1	2020.7.15	2020.8.1	2020.8.15	2020.9.1
1	read data	2020/6/25	2020/7/1	6 days	■					
2	user function	2020/7/1	2020/8/15	45 days		■	■	■		
3	function test	2020/7/7	2020/8/15	38 days		■	■	■		
4	last year objectives	2020/7/7	2020/8/7	30 days		■	■	■		
5	GUI	2020/8/7	2020/9/1	21 days				■	■	
6	integrate test	2020/8/7	2020/9/	30 days				■	■	■

GitHub repository: https://github.com/acse-2019/irp-acse-yz219

Bibliography

[1] Boville, B., and P.R. Gent, 1998: The NCAR Climate System Model, version one. J. Climate, 11, 1115-1130.

[2] Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., Stouffer, R. J., and Taylor, K. E.: Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization, Geosci. Model Dev., 9, 1937-1958, doi:10.5194/gmd-9-1937-2016, 2016.

[3] Gilles Sommeria (former CNRS-WMO, Geneva) and Ludovic Touzé-Peiffer (Laboratoire de Météorologie Dynamique - CNRS, Paris)