

# CMSC6950 Report: MAGICC, a Python wrapper for the simple climate model

Omid Tarkhaneh

June 2021

## Abstract

This report demonstrates some computational tasks using a python wrapper for the simple climate model called MAGICC. In this regard, we employed some defined scenarios in MAGICC to perform computational tasks in python. The achieved results are visualized, and the results are analyzed based on the plots obtained from the computational tasks.

## 1 Introduction

Pymagicc is a Python interface for MAGICC, a reduced-complexity climatic carbon cycle model written in Fortran [1, 2].

MAGICC (Model for the Assessment of Greenhouse Gas Induced Climate Change) is commonly used in climate policy evaluations to predict future emissions trajectories [1].

This report uses some scenarios defined in the MAGICC called Representative Concentration Pathway (RCP). The Representative Concentration Pathway (RCP) is a greenhouse gas concentration (rather than emissions) trajectory. For the IPCC's fifth Assessment Report (AR5) in 2014, four paths were employed for climate modeling and research [3]. The RCPs are named after a possible range of radioactive forcing levels in the year 2100 (formerly RCP2.6, RCP4.5, RCP6, and RCP8.5).

In what follows, we first define the project and the used software to run the project, and then it discusses the computational tasks, results and analyzes the visualized results. Finally, the conclusion will be drawn.

## 2 Project topic and used software

In the project, we visualized the mean global temperature, the atmospheric concentration of  $CO_2$ , and emission of  $CO_2$ , fossil and industrial based on the generated data in different scenarios (RCP26, RCP45, RCP60, and RCP85) provided by the Pymagic wrapper.

This project uses Python 3.8 and the PYMAGIC wrapper to produce data and perform the visualization. In Pymagic, there are built-in plotting functions that we have not utilized in our project; instead, we were interested in using the Matplotlib library in python, which is a standard library for the plotting issues.

In the project, there are four python scripts, each performing different tasks. In LoadData.py, after doing some computational tasks in the “run” method, which is a built-in function provided by the Pymagic, some data are generated and saved in terms of CSV file format used by other python scripts for generating visualization. Other scripts files such as Atmospheric.py, Emission.py, and Stemperature.py will read the generated CSV dataset and used Pandas library in python to convert the dataset into a Dataframe which ease the working with the data for us.

After performing some preprocessing on the data, such as filtering, dropping Nan values, and dropping some unnecessary columns, data are provided to be visualized using the matplotlib library in python. To run the project, it is provided with a “make” file that helps the project to be user-friendly where it can be easily used to generate desired data and visualization. If users are going to run the code without using the make file, they should first run the LoadData.py file to generate the datasets and then run each of the files mentioned above to generate visualizations, which are generated in terms of .png files. The instructions to run the project and see the visualization is provided in our Github repository <sup>1</sup>.

## 3 Computational Tasks

This section will introduce the “run” method in the Pymagic library. To know what computational tasks are performed inside this method, we have analyzed the source code of this method. According to the source code, this method will perform parsing the data and returns MAGICC Data object

---

<sup>1</sup><https://github.com/OmidTarkhaneh/CMSC6950project.git>

containing that data in its "df" attribute and metadata and parameters (depending on the value of "include parameters") in its "metadata" attribute. We performed more tasks on the returned data, which are described in the following subsections. We have also performed some computation on a raw dataset without using the "run" method in the Pymagic library and try to manipulate datasets based on our options.

### **3.1 Generate the surface temperature**

Different employed scenarios in MAGICC have some features and samples. There are various features in the dataset like region, scenario, variable, and so forth.

We have collected the results of all scenarios (RCP26, RCP45, RCP60, and RCP85) in a CSV file for this task. After collecting the data into a CSV file, different computational tasks have been performed, such as dropping duplicate values, Nan values, etc. In addition, one feature of dataset is related to the region, and for predicting the surface temperature, we have limited the dataset to only count the world region despite some other regions in the dataset. Moreover, we have filtered the variable to surface temperature. In the dataset, variable attribute contains different values related to the greenhouse gases, surface temperature, emissions, etc. Given the achieved dataset after performing some computational tasks on it, we tried to visualize the obtained data to predict mean surface temperature. In figure 1, we have depicted results of the data for all the employed scenarios for surface temperature in the world.

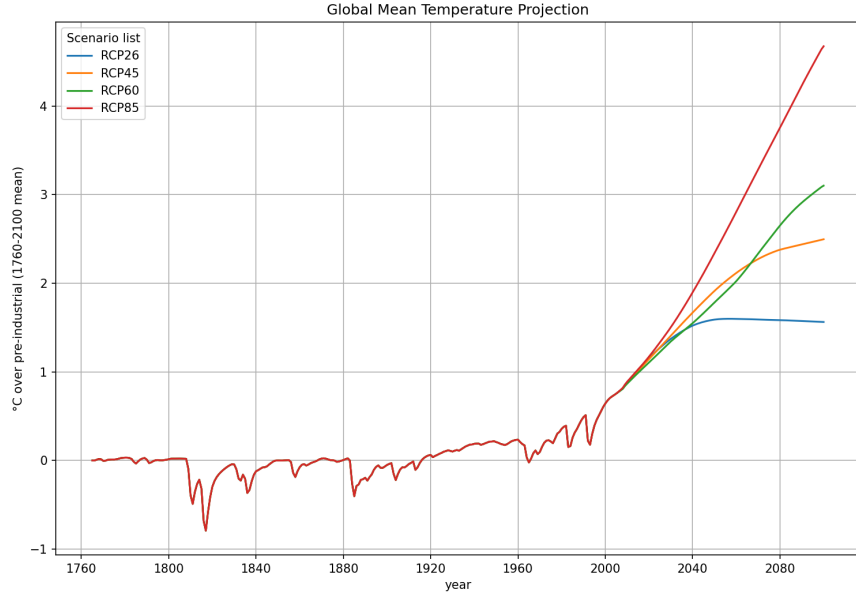


Figure 1: Mean surface temperature based on employed scenarios.

Figure 1 depicted the surface temperature from 1760 to 2100 years. The depicted figure shows an increasing trend in all the scenarios, although there are some fluctuations in some years. Since 2000, we have a much more increasing trend where the surface temperature experienced the highest value in scenario RCP85.

### 3.2 Generate the atmospheric concentration (CO2)

This section evaluates the atmospheric concentration of CO2 and utilized different scenarios of RCP26, RCP45, RCP60, and RCP85. To do so, first, we have performed some computational tasks using a "run" method in the Pymagicc library in python. The mentioned method does some computation such as parsing the raw dataset. Then, we will filter the dataset based on region and variable attributes where we have used "world" for the region and "atmospheric concentration—CO2" as the variable in the dataset. After creating our dataset, we have visualized the data to evaluate the amount of atmospheric concentration between the years 1760 to 2100 in the world.

Figure 2 depicted the obtained data and shows the atmospheric concentration in different years.

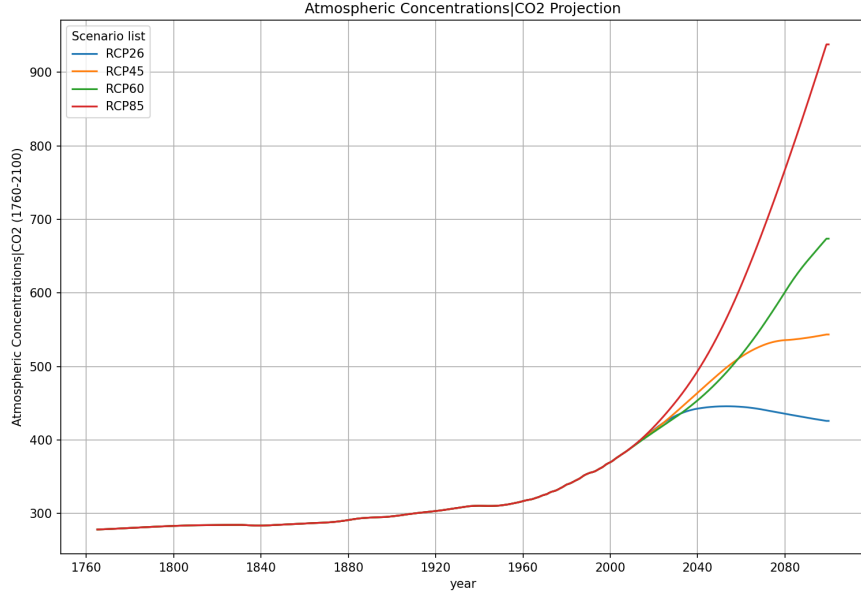


Figure 2: Atmospheric concentration based on employed scenarios.

Figure 2 shows the atmospheric concentration of CO2 from the years 1760 to 2100 in the world. According to the figure, from 1760 to 1960, there is not a considerable increment in terms of atmospheric concentration of CO2; however, since 2000, especially the world experience a huge concentration where its highest value is more dominant in scenario RCP85. To the best of our knowledge, this sharp increment is due to the industrial improvement which is created by mankind. After the industrial revolution, generating greenhouse gases increased, and this is one main reason for increasing the surface temperature too.

### 3.3 Generate Emissions of CO2, Fossil, and Industrial

In this section, we worked on a raw dataset where we have not utilized the run method in the Pymagic library in python; rather, we have tried

to handle the data and manipulate it ourselves. The preprocessing that is done includes dropping Nan values and dropping duplicate samples, filtering datasets based on region and variable, dropping some of the features that are not needed to make the projection, etc. After doing preprocessing, the dataset was visualized, and it shows an interesting diagram of emission of fossil and industrial as well as CO2 in the world. This time we tried to depict the data from 2000 to 2100. Figure 3 shows the results.

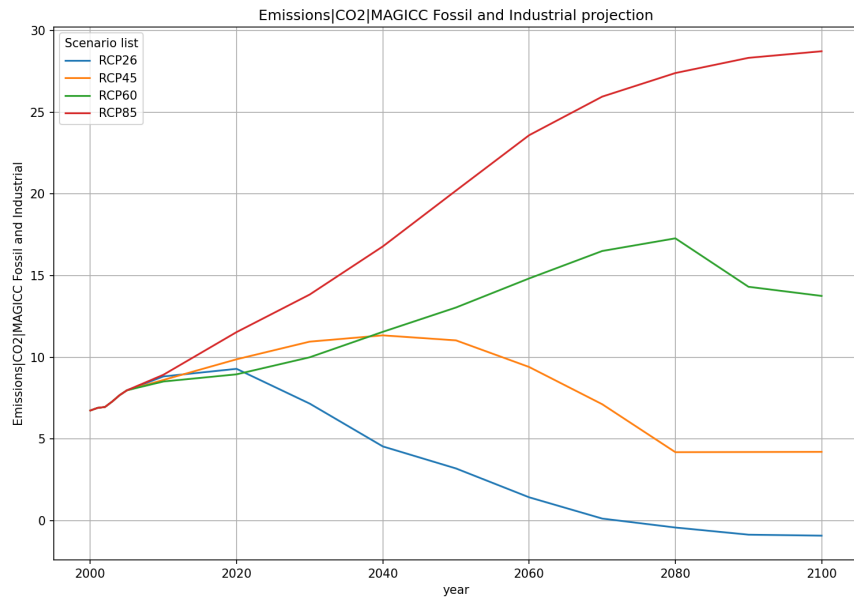


Figure 3: Emissions of CO2, Fossil, and Industrial based on employed scenarios.

In Figure 3 there are different trends for each scenarios. Scenario RCP85 shows an increase in the emission of the selected parameters (fossils and industrial, and CO2); however, scenarios such as RCP26 and RCP45 show a decrease in the emission of the selected parameters. The interesting thing is that from the years 2050 to 2100, there is no such increase in emissions of the selected metrics, and even in some scenarios, the emission value is decreased.

## 4 Conclusion

This report visualized some data based on some scenarios utilizing the Pymagic library in python. The results show that we will see an increasing trend in terms of atmospheric concentration, emission of greenhouse gases, and ultimately the surface temperature in the future. One exciting thing was that from the years 2050 to 2100, we might not experience a sharp increase, and we may even see a decrease base on some scenarios.

## References

- [1] Gieseke, Robert, Sven N. Willner, and Matthias Mengel. "Pymagicc: A Python wrapper for the simple climate model MAGICC." *Journal of Open Source Software* 3, no. 22 (2018): 516.
- [2] Meinshausen, Malte, Sarah CB Raper, and Tom ML Wigley. "Emulating coupled atmosphere-ocean and carbon cycle models with a simpler model, MAGICC6–Part 1: Model description and calibration." *Atmospheric Chemistry and Physics* 11, no. 4 (2011): 1417-1456.
- [3] Moss, Richard H., Mustafa Babiker, Sander Brinkman, Eduardo Calvo, Tim Carter, James A. Edmonds, Ismail Elgizouli et al. "Towards new scenarios for analysis of emissions, climate change, impacts, and response strategies." (2008).