

# Graphing CO<sub>2</sub> Concentration Over Time

Last modified: September 4, 2020

## Python Program

The graphing program is used to generate graphs which depict the rise and fall of carbon dioxide (CO<sub>2</sub>) concentrations over time for a particular cell. The user specifies a cell to generate a graph for using the graphical user interface (GUI) and the program will generate a visual representation of the data. The program requires that a Cadmium output file be specified in order to create any graphs. Both of Cadmium's output files ("output\_messages.txt" and "state.txt") can be used to generate graphs.

The name of the output file can either be specified on the command line before the GUI is launched or by selecting a file using the provided file selector within the interface.

There are two modes for the program: transient and store. Transient mode will only load what is required for a specific set of coordinates and will only access the file after a graph-generation request is made. This functionality means that the program will have a quick initial load time but will take longer to generate graphs.

Store mode will parse the entire file upon the program's launch and will retain the required information in memory. For small to medium files, this process can take anywhere from thirty seconds to several minutes. Large files may pose issues as the time required to load can become excessively long and the program may run out of memory. The benefit of this mode is that once the data is parsed and loaded, generating graphs is relatively fast as the program simply needs to do a dictionary search to find the necessary data.

The mode in which the program runs must be set on the command line. By default, the program will run in transient mode.

## Command Line Usage

The program uses two modules that are not typically included in the default installation of Python 3. The two modules are "pandas" and "plotly.express". The libraries can be installed using:

```
pip3 install pandas
pip3 install plotly==4.9.0
```

Should any issues arise using pip3, installation instructions for Plotly can be found here:

<https://plotly.com/python/getting-started/>

The program has the following usage:

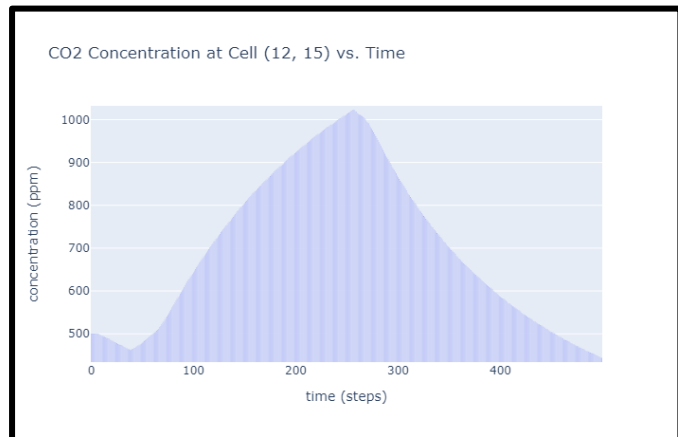
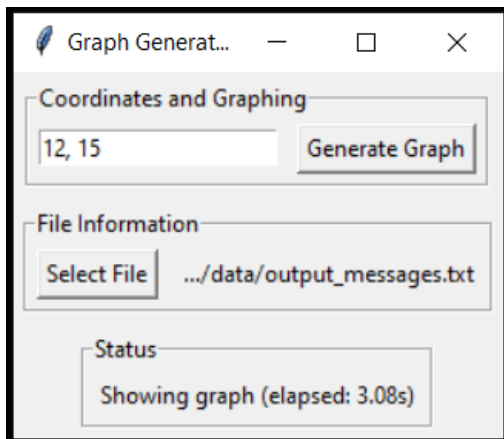
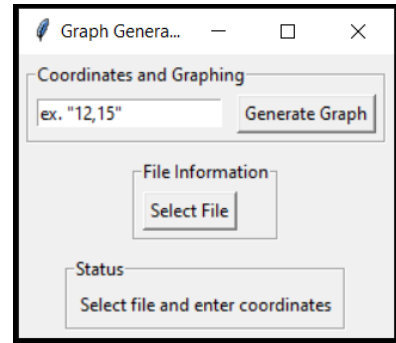
```
createGraph.py [-h] [--filename FILENAME] [--store]
```

## Using the Interface

The GUI uses Python's tkinter which should be installed with most Python 3 installations. Below are the steps to generating a graph using the GUI.

1. If no file was specified on the command line, click the "Select File" button and pick the Cadmium output file that contains the data which should be graphed. If the program is in store mode, there will be a waiting period where the program must access the file and parse the data. The time taken increases with the file size.
2. Clear the text field in the "Coordinates and Graphing" section of the interface and enter the coordinates that should be plotted. Examples of valid formats are "12, 15" and "12,15" (without quotation marks). Do not include any other punctuation. Note that the dimension of the given coordinates and the dimension of the selected file's scenario must match in order for a graph to be created.
3. Click "Generate Graph". If the program is in transient mode, there will be a waiting period where the program must access the file and parse the data. The time taken increases with the file size.
4. A new tab will open in a browser which will contain an interactive graph. In the event that the tab loads indefinitely, close the tab and click "Generate Graph" again.

In order to exit the program, simply hit the "X" at the top-right of the window.



## Changing Models

In the event that the program is used for other models, this program cannot be used without significant modification. The primary change that would need to occur is a result of the fact that the program expects coordinates as input. Given something that is not a coordinate, the program will display an error and wait for input in the expected format. Aesthetically, the labels on the graphs that are generated would need to be changed. These can be changed in the file “Constants.py”.

```
10     # Class: Graph
11     # Purpose: graph constants
12     # Arguments:
13     #     none
14     class Graph:
15
16         labelX = "time (steps)"
17         labelY = "concentration (ppm)"
18         title = None # set to None for Graph.getTitle to create a title
```

In the image, the labels are set on lines 16, 17, and 18. If the title is set to *None*, the function “Graph.getTitle” in “Graph.py” will generate a title.

The program expects to find data of the format:

`*(NUM, NUM, NUM) ; <NUM,NUM,NUM>*`

where the parentheses (round brackets) are the first pair from the left of the line and the angled brackets are the first pair from the right of the line. The values within the parentheses represent the coordinates (or cell\_id) of the cell. For the angled brackets, the program only uses the second value while the first and third values are discarded. This focus on the second value means that it is possible to have as few as two (or more) values within the angled brackets. For the CO<sub>2</sub> model, the program expects to find the concentration of the gas in this place. If another model is to be graphed, the logs either need to match the expected format or the program will need to be modified. How the files are parsed is handled in “Parse.py”.

The program’s code may require changes that have not been documented here.