

# **A GUI for POLYRATE**

Akilan Rajamani, Maxime Infuso, Denis Duflot and Celine Toubin

*Univ. Lille, CNRS, UMR 8523 - PhLAM - Physique des Lasers Atomes et Molécules, F-59000 Lille, France*

## **Licensing and copyrights**

The code is written only for academic purpose and free of cost.

## 1. Installation

The Graphical User Interface (GUI) is written in Python 3.9 with windows OS and the efficiency of the code is not checked with other versions or other operating systems. But it is understandable that the code may works well with any operating systems having python versions later than 3.9. In order to use the code, user have to install the following python libraries,

**A. Tkinter**

**B. Pandas**

**Here are the internet links for the corresponding python package:**

**Tkinter:** <https://tkdocs.com/tutorial/install.html>

**Pandas:** <https://pandas.pydata.org>

I highly recommend you to download the Anaconda package that provides you all the environment to code and execute Python scripts. In addition, I recommend that you also install the corresponding packages through this environment, using the command:

**conda install [name\_of\_the\_package]**

However, you can install all the packages using pip install if you do not want the Anaconda package.

**To test if your installation is correct, open a terminal and write:**

~~~

**python**

~~~

This will open a local python environment in a terminal. You can verify the version of python at this step. Try to “import” the modules to see if it works:

~~~

**>>> import [name\_of the package] #Generic name**

**>>> Import pandas #Example with the pandas module**

~~~

If there is no message and the prompt is appearing, the package is well installed. If you receive a “ModuleNotFoundError”, you probably have installation issues with the package. If it is the case, try checking your python path:

~~~

**which python**

~~~

**In a bash shell, and check if the resulting path is correct.**

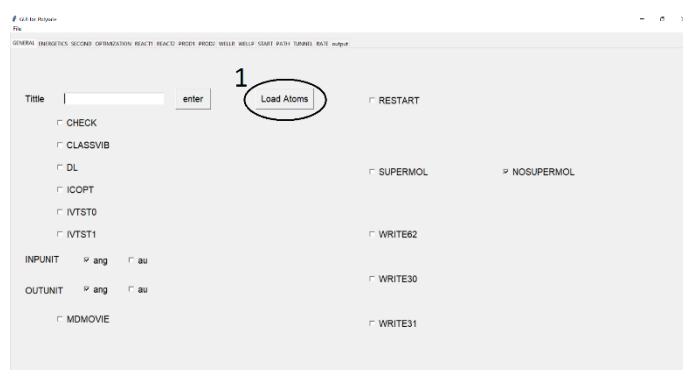
**Finally, I think that the installation is the same regarding Windows/Linux/MacOs. Only the steps to install Python and the corresponding packages are different, but all well explained in the corresponding websites.**

## 2. Tutorial

This code is written for beginners and intermediates to create input for Polyrate from Gaussian outputs. The keywords used in the code are from Polyrate and users can read the Polyrate manual “[https://comp.chem.umn.edu/polyrate/210719\\_2000\\_Polyrate\\_17-C\\_Manual.pdf](https://comp.chem.umn.edu/polyrate/210719_2000_Polyrate_17-C_Manual.pdf)” to understand the keywords before using them. Since, this code is written for beginners and intermediates, frequently used keywords are only added to this code. If users want to use other keywords, they can manually add them. Here, a simple example is shown to create Polyrate input.

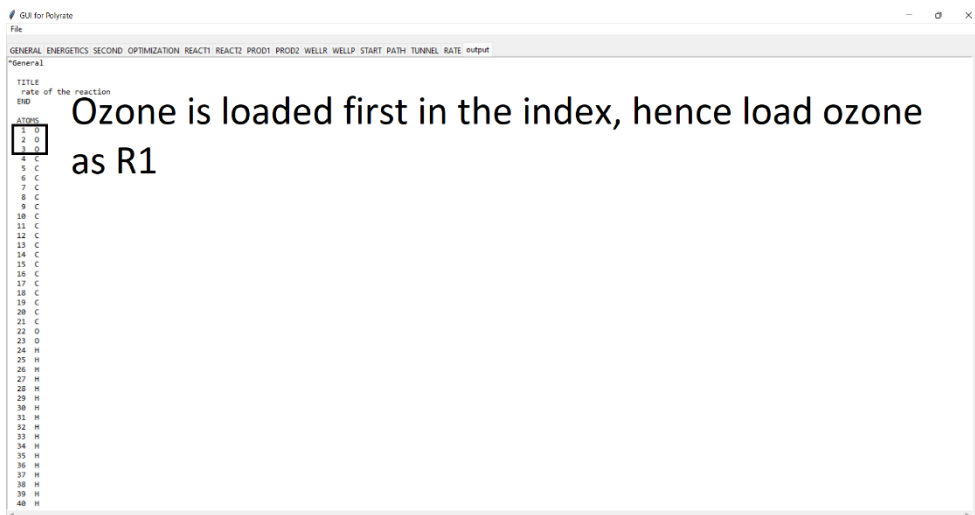
Step 1: Load the code in your terminal using Python. [example: “python main.py” or “python GUI.py”]

Step 2: load your TS file (example “ts.out” or “ts.log”) in load atoms. Then Title of the work. Based on your needs, key words can be selected from the General section.



Step 3: In order to choose appropriate keywords in ENERGISTICS, SECOND, OPTIMIZATION sections, please check Polyrate manual “[https://comp.chem.umn.edu/polyrate/210719\\_2000\\_Polyrate\\_17-C\\_Manual.pdf](https://comp.chem.umn.edu/polyrate/210719_2000_Polyrate_17-C_Manual.pdf)”.

Step 4: Before loading Reactant one, go to the output and check the indexes. For example, in Oleic acid + Ozone reaction, ozone is loaded first atom indexes, hence load ozone as Reactant one.



Ozone is loaded first in the index, hence load ozone as R1

Then Oleic acid as reactant 2. Every time you provide an input, do not forget to press “Enter”.

Step 5: In REACT1, REACT2, PROD1, PROD2, START, WELLR, WELLP, load the gaussian outputs first before using STATUS keyword to avoid any errors.

Step 6: In RATE section, multiple activation energy pairs can be added.

We strongly recommend the users to understand the keywords and their usage from Polyrate manual [https://comp.chem.umn.edu/polyrate/210719\\_2000\\_Polyrate\\_17-C\\_Manual.pdf](https://comp.chem.umn.edu/polyrate/210719_2000_Polyrate_17-C_Manual.pdf) to effectively use this code.

Step 7: Finally, go to “output”, check and copy your Polyrate input, save it as “.dat” file using notepad or any text applications for Polyrate calculations.

We have also added a “potential.sh” file which is written in bash and python to create a potential40 file. The script is working well in Linux and Mac. Windows users can use this script with Cygwin or Linux terminal from Microsoft terminals but the working of script is not checked with windows system.