TOPAS Automator User Guide

Introduction

TOPAS Automator is - so far - a tool to help working exclusively with simulations regarding neutron resistive plate chambers (nRPCs) in the TOPAS MC toolkit. The program provides different setups, but all of them in the spectrum of detecting neutrons through resistive plate chamber for neutrons.

The one and only purpose of this program is to automate and simplify the creation of the inputs you are going to run with TOPAS. No more writing endless .txt files! With TOPAS Automator, you just make your way through the interface and it creates the file you will need to run your simulation in TOPAS MC.

About

TOPAS Automator was created as part of one of many LIP internships at the NUC-RIA group, in the year of 2023. Carolina Felgueiras (M.Sc. at the time) was working with the detection of neutrons using resistive plate chambers and was running her simulations in the TOPAS framework. Her work required to run several simulations with different setups and this internship came as a way to help her automize this process.

And so, entered the internee Tomás Campante (B.Sc. at the time) who helped Carolina in her work during the summer months of 2023, by creating TOPAS Automator interface.

Download & Installation

Click here to access TOPAS Automator Download page.

Before running the topasautomator.py file with python3, make sure you have the Tkinter and Pillow modules from Python in your machine.

To do so, if you are on Windows or Linux, open your terminal and type in (one line at a time).

```
python get-pip.py

pip install Pillow
pip install tk
```

If you are on macOS and get some kind of error using the pip command, try this one instead:

```
pip3 install Pillow
brew install python-tk
```

If you are on Linux and still get some kind of error, you'll want to try running the following lines in your terminal.

```
sudo apt-get install python3-tk
sudo apt-get install python3-pil python3-pil.imagetk
```

If you are experiencing any more difficulties, we encourage you to send an email to ***example@example.ex*** .

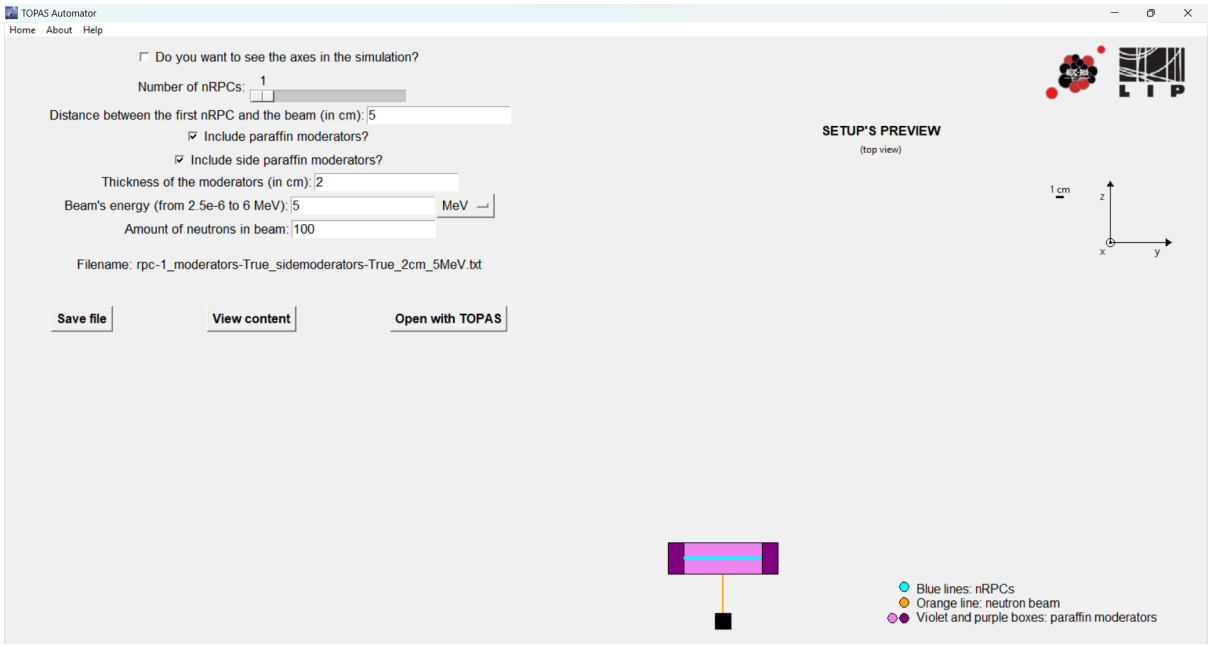
Getting started

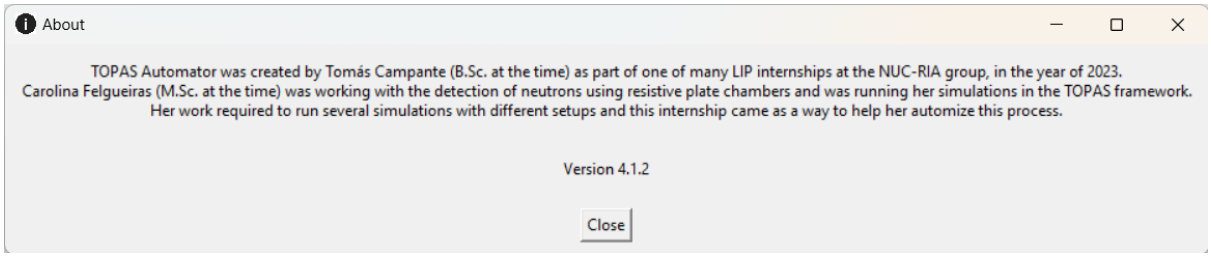
To run topasautomator.py with all of its functionalities, we advise you to open it with your terminal (Visual Studio Code usually works too). To do so, you just have to open your terminal, make your way to the directory where the downloaded folder is (make sure it is unzipped) and then type

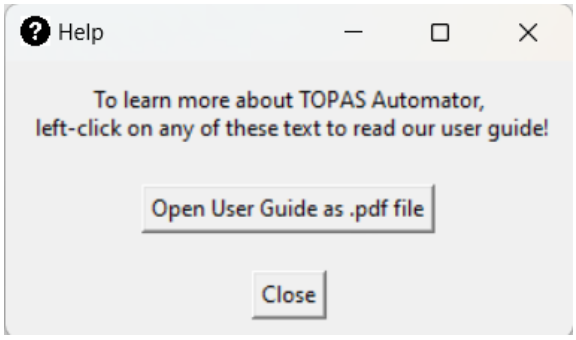
```
python3 topasautomator.py
```

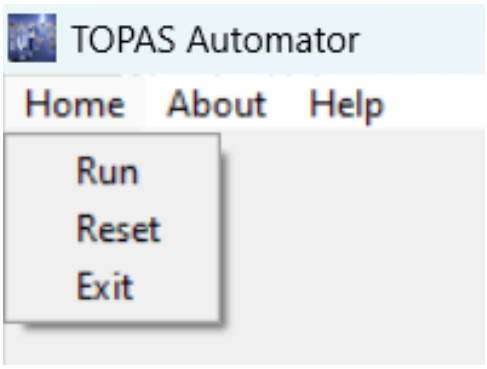
There, it will open a window where you can fill the entries and the .txt file with the features you want will be created, and if you click in the **Open with TOPAS** button it saves the file you just generated and opens it with TOPAS MC.

Overview









Analyzing more thoroughly the program's window, you can see at the left top corner three buttons:

- Home: where you can run (equivalent to Open with TOPAS), reset and exit the program;
- About: where you can learn more about the creation of the TOPAS Automator;
- Help: where you will find your way to this user guide.

When our eyes start to wonder around the window, we see the drop-downs menus, entries and buttons where you select, click and type according to what you wish to do.

The "View content" and "Open with TOPAS" buttons also save the .txt file by default. If there is already a file with that name in the directory, it will be overwritten.

At the right, you'll see geometric figures. This is a preview of the setup you will see with TOPAS, when you open the file you are generating. It is a previsualization of the simulation's setup.

If the image following this indications is slightly different than yours, do not worry, it must be due to the operative system or version difference.

Going in depth

In this section we will go in depth on how the program is built and on how it works exactly.

Regarding the creation of the window and everything that is clickable, it was used the python library Tkinter. This is a library frequently used when it's pretended to build interfaces and GUIs in python.

Regarding the creation of the file: the folder where topasautomator.py is also has another folder (DoNotDelete) with some .png and 6 more very important text files - this folder (along its content) should not, under any circumstances, be deleted or renamed. Those files contain (when put together) the necessary code to run the simulation for 1 nRPC (with specific characteristics). After the user selects and inputs the parameters' values that are relevant to change in the experiment, the respective variables are searched for in those files and are swapped accordingly.

The program is built to simulate the detection of neutrons with nRPCs. It supports this kind of simulation with moderators and without moderators and both options are accounted for in the interface.

If you need to change some parameters, e.g. the shape of the beam, you can just access those .txt files and search for that parameter by hand. Then you save them and when you will use the interface again that change will be present.

If you wish to use the program to simulate other stuff, this documentation does not exclude the reading and understanding of the code.

FAQs and troubleshooting

The automator froze after I close TOPAS.

Unfortunately it seems to be common for the python program to freeze after running TOPAS, running the simulation and closing TOPAS. Although we are not sure about the origin of this problem, we believe this happens due to shortage of available RAM.

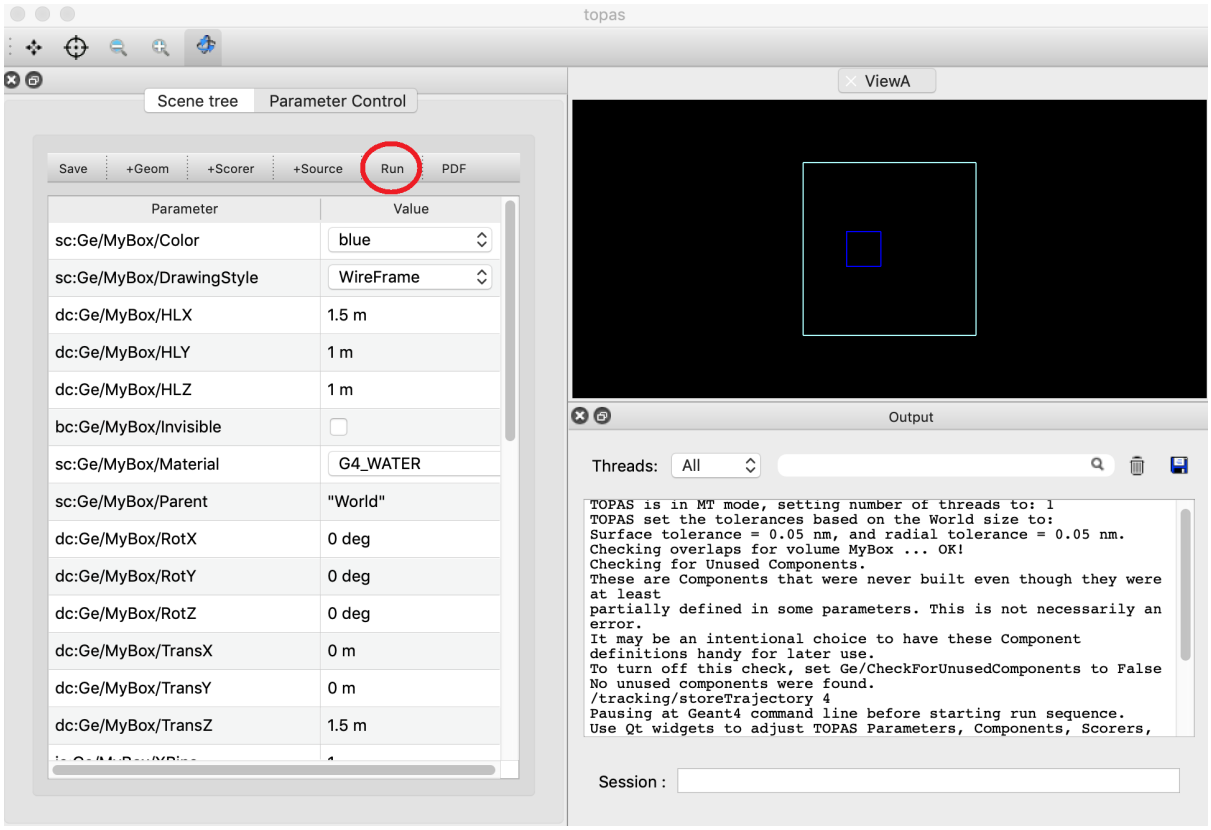
We advise the user to just CTRL+C on the terminal (to terminate the currently-running program) and re-run the bash command to re-open a new interface. Probably the old interface will stay open (but frozen), however a new one will appear and you can just work on that one. In case it does not, you just close the terminal and repeat the process that led you here. Also, you can just use the topasautomator.py to create the .txt file automatically and run the created files on TOPAS by your self through the topas alias on the terminal.

```
topas filename_example.txt
```

Once TOPAS is open, how do I run the simulation?

This is more of a TOPAS related question than related to the automator himself, but we are always happy to help.

If TOPAS' windows popped up and you can see the setup for the simulation, you go to the top-left side of TOPAS' window and click run.



You can see that it is running because in the terminal you will see new lines appearing. After all the events run (1 neutron in the beam = 1 event) it will take some time but it should reload the setup preview in TOPAS and now you will see a lot of crazy lines with different colors. The yellow lines are the neutrons and the grey lines are other random particles - to learn more go to TOPAS' user guide. Be aware that the more neutrons are in the beam, the longer this process will take, and more amounts of RAM it will consume.

Can you turn off the axes after TOPAS is open?

No, that functionality is not available in TOPAS.

Source

If you for any reason want to have access to the code itself, you just have to open the topasautomator.py and auxialiaryFunctions.py with your regular .py file editor.

The code is duly commented, you will find your way in it.

Acknowledgements

I would like to thank my supervisors Carolina Felgueiras and Daniel Galaviz for guiding me through this process and being always available to clarify my questions.

Also, a huge thanks to all of the NUC-RIA group members and fellow internees for providing an excellent work environment.

Thanks, mom! <3

Last modified: 29 august 2023

Tomás Campante