# 5-Axis 3D Printer Software User Manual

## 

Authors:

Connor Wyatt

CJ Young

Submitted in fulfillment of the requirements for COMP 4710 Senior Design to the

Department of Computer Science and Software Engineering, Samuel Ginn College of

Engineering, Auburn University

Auburn, Alabama

December 4, 2023

## Comp 4710 Senior Design Project

## Fall 2023

## 

## Table of Contents

[**Table of Contents**](#_b7x1mj5khr0g) **ii**

[1. System Metaphor](#_oyf70wf85ax5) 1

[2. System Overview](#_f096vfqr61g) 2

[3. Set-up](#_za4zybp0gwlp) 3

[4. Overall Process](#_5u0ogza1f6qc) 4

[5. Slicer/Simulation Process](#_36ez2h4zb7ah) 6

[6. G-Code Translation Process](#_bkxtbyhkkka3) 7

### 

### 

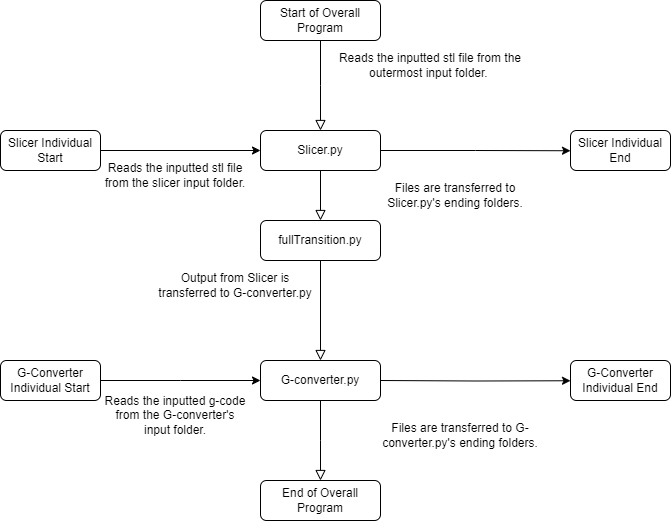
### 1. System Metaphor

3D Printers are now widely known for their ability to replicate plastic objects by utilizing instructions known as G-Code. However, a standard 3D Printer suffers in a few fields: Larger models require numerous supports, general printers can have a difficult time producing specific shapes, and many shapes are simply infeasible due to the layout of the printer.

The 5-Axis 3D Printer provides a solution to these issues by utilizing a main rotatable support instead of a standard 3-Axis printer. However, this introduces an additional need: the G-Code utilized for standard printers does not correctly translate to the coordinates needed for the 5-Axis printer. This software provides that exact solution, and, by following this manual, you can utilize it to print your very own 3D models. [taken from Cycle 3 Report, Fall 2023]

### 

### 2. System Overview



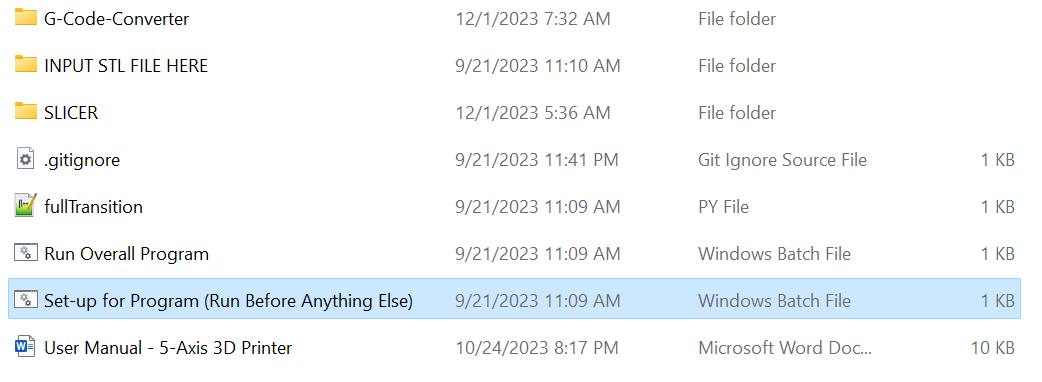
*Figure 1 Flow of the System [taken from Cycle 3 Report, Fall 2023]*

The entire program works in a set number of modules. This system can be run at a number of different starting points, depending on what actions the user needs to perform. Starting points in the process will be noted in their respective sections.

The program also works off of running a number of files and folders. Any section can be started utilizing its “Run (Program Segment)” batch file, and requires that the user input certain files into their respective folders. Visual indicators of these locations are provided.

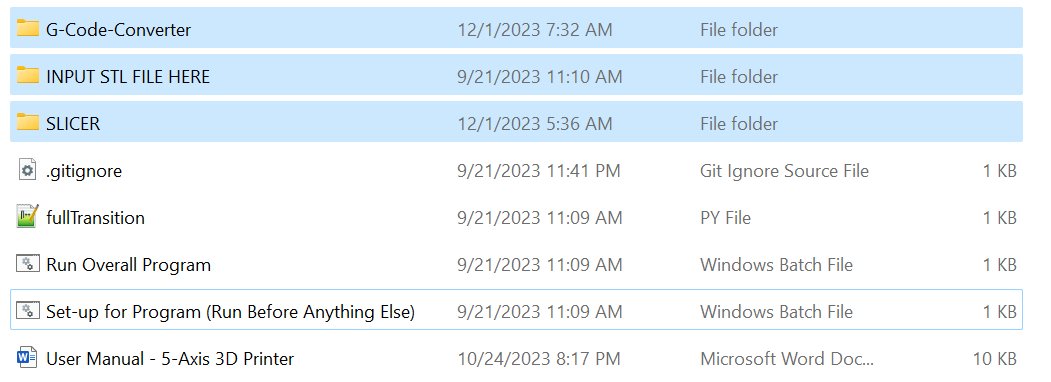
### 3. Set-up

1. Before utilizing this program, you will want to install python on your computer.   
   Python installations can be found at <https://www.python.org/>, no particular version is required, using the most up-to-date version should suffice.
2. Run the “Set-up for Program” file found in the main folder. This will automatically install all necessary components needed to run the remaining program.



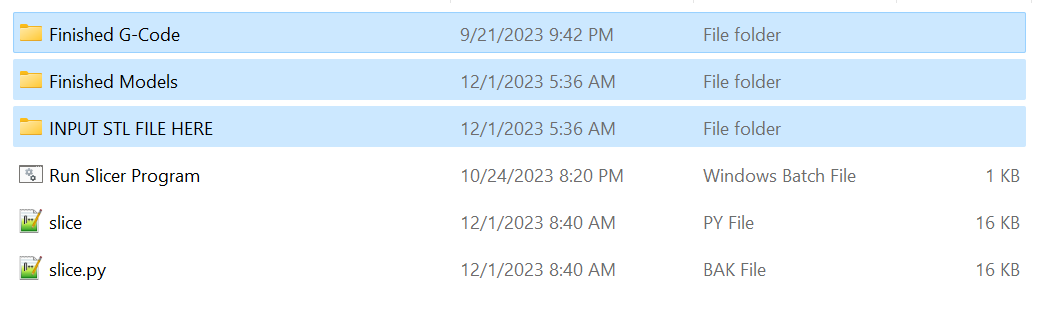
*Figure 2 Set-up for Program*

1. TROUBLESHOOTING:
   1. If your program will not run after these steps, ensure that the following folders are present:
      1. “INPUT STL FILE HERE” in the general files.



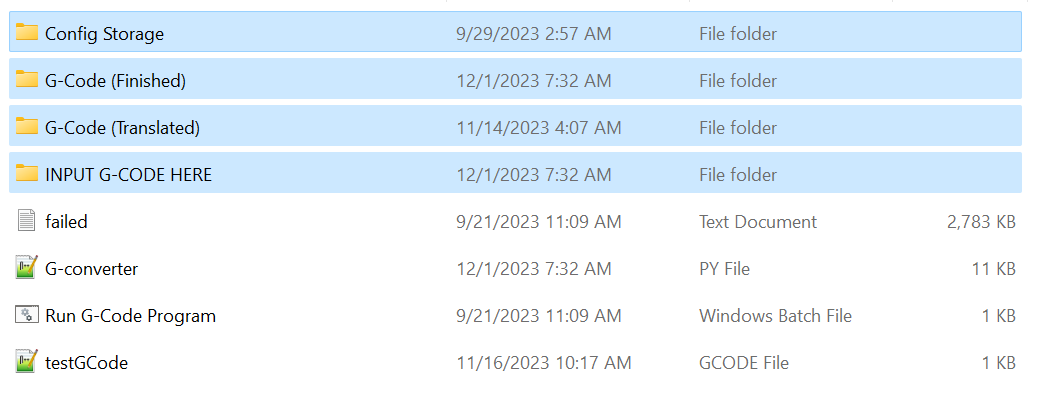
*Figure 3 Important Folders (General)*

* + 1. “Finished G-Code”, “Finished Models”, “INPUT STL FILE HERE” in SLICER.



*Figure 4 Important Files (Slicer)*

* + 1. “G-Code (Finished)”, “G-Code (Translated)”, “INPUT G-CODE HERE”, and “Config Storage” in G-Code-Converter.



*Figure 5 Important Files (G-Converter)*

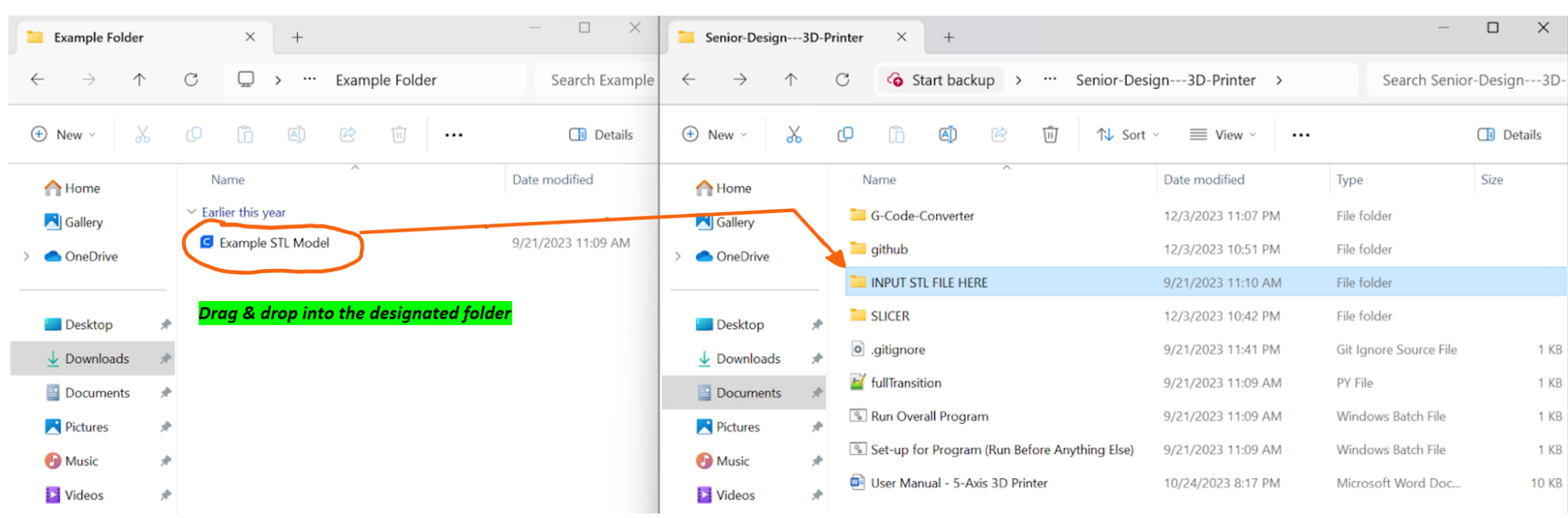
* 1. If your stl files are causing errors when running the program, consider reimporting them or port them from a different program. Certain programs will port the stl file differently. We specifically utilized Ultimaker Cura to export any utilized STL files.

### 

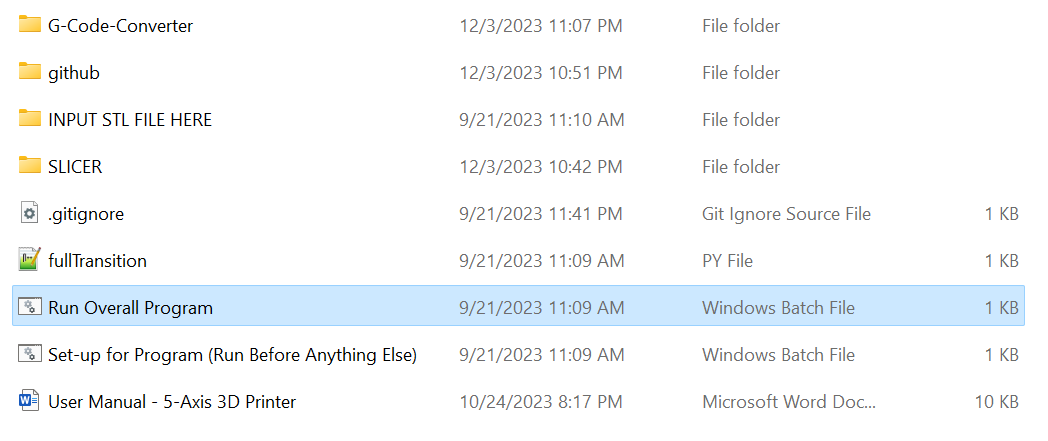
### 4. Overall Process

Now that you have all of the required files, you may now utilize the entire program.

1. On the file level of “fullTransition.py” and “Run Overall Program”, insert any STL File into the folder “INPUT STL FILE HERE”.   
   The program can handle more than one of these files being in the input folders, but it will only handle one stl file per run. Thus, it’s recommended to utilize only one file when running any segment.

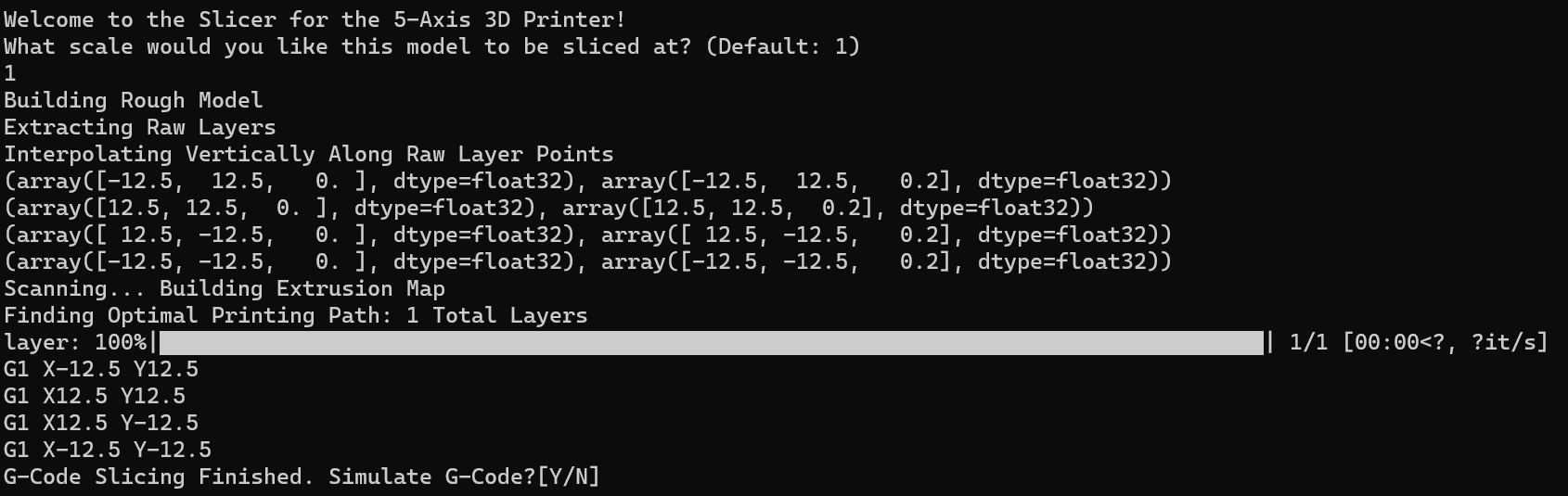
  
 *Figure 6 How to Input STL Files*

1. You can now run the batch file labeled “Run Overall Program” by double clicking it, and a command prompt should open. The program will automatically start the “Slicer/Simulation” process.

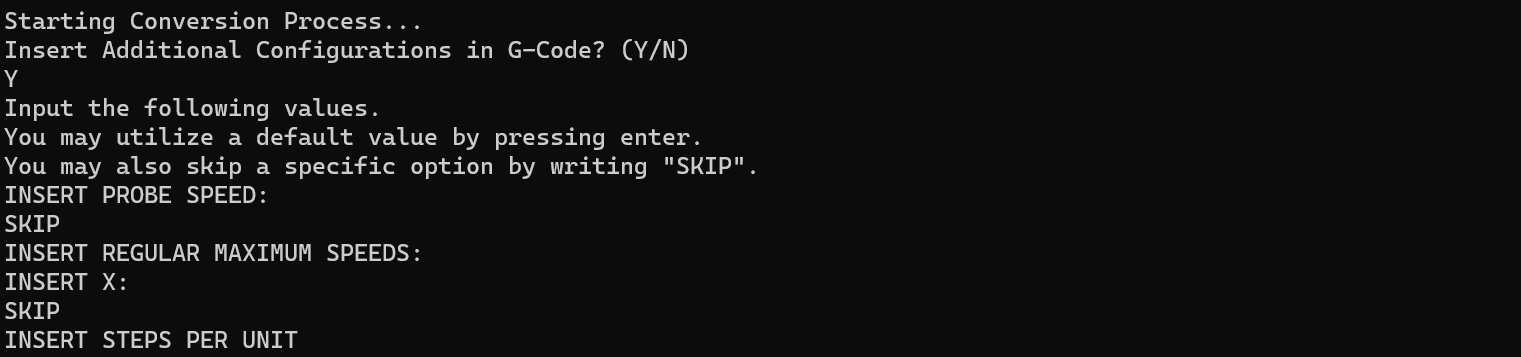


*Figure 7 Run Overall Program*

1. After this process concludes, the program will utilize a utility script to move the finished G-Code over to the G-Code-Converter file, where it will start the “G-Code Translation” process.



*Figure 8 Slicer Terminal*

**

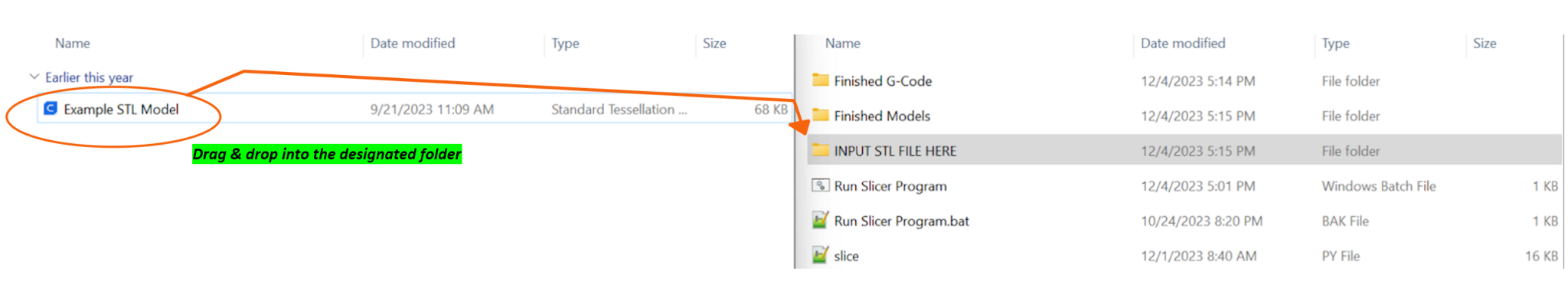
*Figure 9 G-Code Converter Terminal*

1. Once this process has finished, you have now seen a simulation of the sliced STL Model and possess a translated version of its G-Code.

### 

### 5. Slicer/Simulation Process

The following processes can be run individually, but are also utilized during the section “Overall Process”, as referenced above.

1. On the file level of “slice.py” and “Run Slicer Program”, insert any STL File into the folder “INPUT STL FILE HERE”.  
   

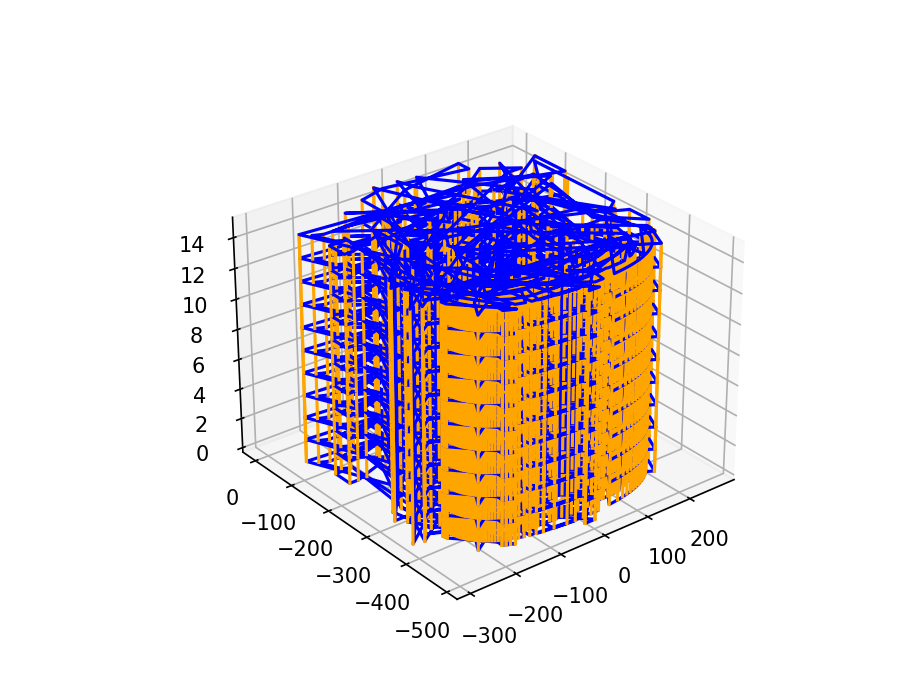
*Figure 10 Slicer Input Example*

1. Run the program by clicking on the “Run Slicer Program” batch file. Once the program has started, you will see an output depicting the model being translated into G-Code inputs. This is an indication that the STL model was successfully read, and also acts as an indication of the simulation length (a larger/complex STL model will take longer to render).



*Figure 11 Slicer Loading Screen*

1. You will then be asked if you’d like to simulate the model. This will give you a look at what the model appears as based on the individual points at each layer. This can be expensive depending on the complexity of the model, but overall is very helpful.
   1. If you answer “Y”, you will then be asked if you’d like to simulate the full model. This will create structures between the layers to give a more accurate version of what will be printed. This is generally up to user taste, as it’s unnecessary for some models, but can be helpful for others.   
      If you do decide to simulate the full model, be aware that this process can lag depending on model complexity, as the system is drawing a large number of lines to simulate the model.



*Figure 12 Slicer Visual Simulation [taken from Cycle 3 Slides, Fall 2023]*

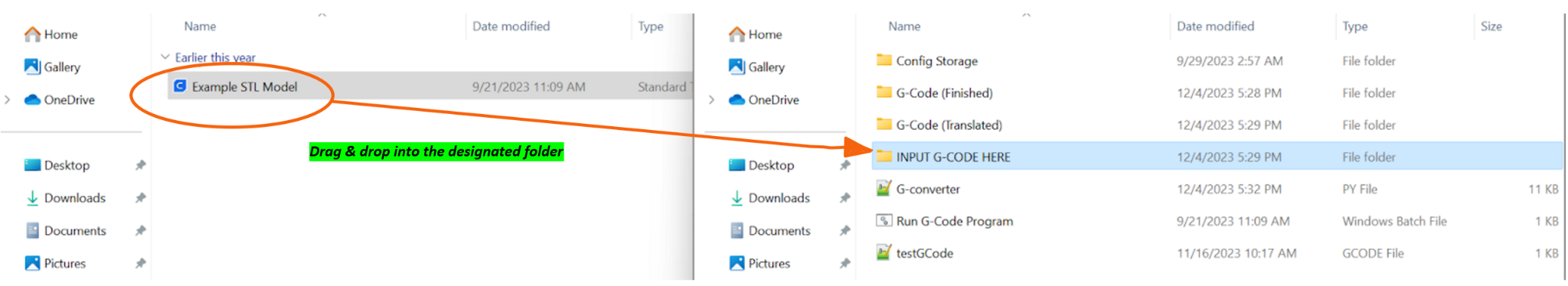
1. Once either choice is made and the process has finished, the process will either close or you will be left with a finished model, with the process prompting you to press any key. Doing so will finish the process and close the model.

### 

### 6. G-Code Translation Process

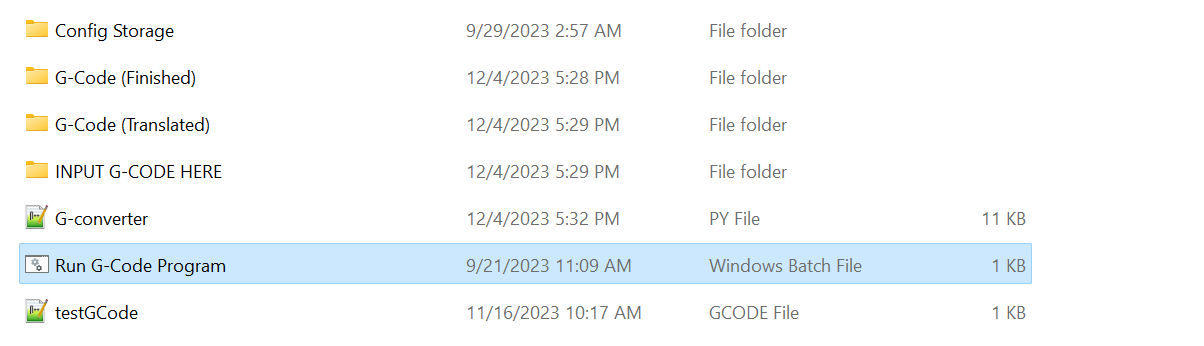
The following processes can be run individually, but are also utilized during the section “Overall Process”, as referenced above.

1. On the file level of “G-converter.py” and “Run G-Code Program”, insert any generalized G-Code into “INPUT G-CODE HERE”.



*Figure 13 G-Code Conversion Input*

1. Run the program by clicking the “Run G-Code Program” batch file. The program should open as a terminal window.



*Figure 14 Run G-Code Conversion*

1. You will then receive the prompt "INSERT ADDITIONAL CONFIGURATIONS IN G-CODE? (Y/N)". Typing “N” will continue to step 4, but typing “Y” will let you use additional specifications, noted in 3a-g. You may also type “SKIP” to avoid putting specific values into your G-Code.
   1. Probe Speed

Helps to determine the precise precision of specific printed items.

Increasing this speed increases the probing speed.

Default Value: 6000

* 1. Regular Maximum Speed

The fastest rate at which the motors can travel.

Faster speeds transmit more kinetic energy to the print, so the print quality can be affected. Slower speeds could be necessary depending on the build of the model.

Default Values: 300, 600, 120, 240, 45, 4

* 1. Steps Per Unit

The number of “steps” between each movement.

Calculates a number of different resources, including the filament printing itself. Modifying this will lead to

Default Values: 25, 5, 25, 2.381, 1, 6, 1

* 1. Maximum Instantaneous Speed Changes (mm/min)

When a stepper motor changes too quickly, it can end up skipping steps.

Changing this value will change the stability of the system.

Default Values: 200, 200, 20, 20, 20, 120

* 1. Fan Options

Sets fan power for each fan.

Ranges from 0 to 255

Default Value: 255, 255, 255

* 1. Temperature Limit

The limit for the heater section of the printer.

This may need to be changed depending on the utilized filament.

This also modifies the general range of the heater sections, which should be monitored closely.

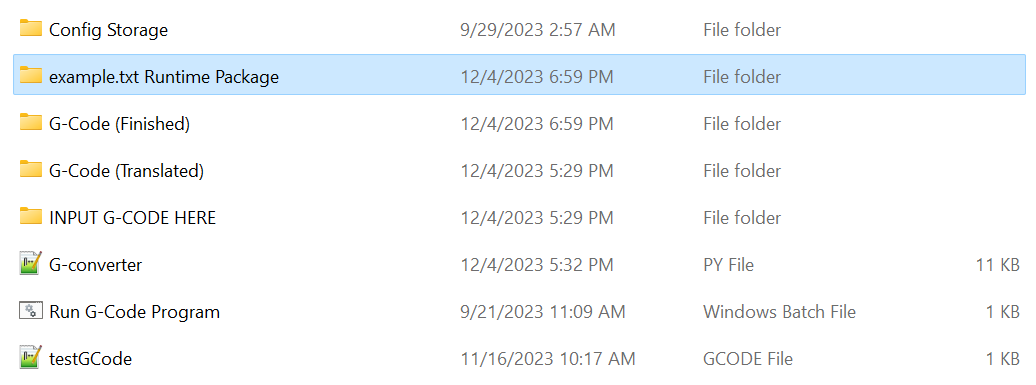
Default Value: 262, 300

* 1. Idle Timeout (seconds)

Sets the time before motors are disabled. General value is set low, so adjust to fit your own time.  
Default Value: 30

Recommended Value (12 hours): 43200

1. At this point, you will be prompted on if you’d like to package your G-Code or not. Typing “Y” will produce an individual file complete with config folder, whereas typing “N” will end the program. Either way, your G-Code will have been translated and placed into its respective folder (the G-Code (Translated) folder if you typed “N”).

****

*Figure 15 Example of Config File*